LEVEL II

(12)

AD A069384

(14) RADC-TR-78-224

In-House Report

(11) April 1979

(13) 123p.

(6) A DESIGN GUIDE FOR BUILT-IN-TEST (BIT)

(10) Anthony Coppola

(16) 2338

(17) 02







ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441



A DESIGN GUIDE FOR BUILT-IN-TEST (BIT)

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-78-224 has been reviewed and is approved for publication.

APPROVED: D F Barber

DAVID F. BARBER
Chief, Reliability Branch
Reliability & Compatibility Division

APPROVED: Joseph J Naresky

JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBRT), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1 REPORT NUMBER<br>RADC-78-224 | 2 GOVT ACCESSION NO. | 3 RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>A DESIGN GUIDE FOR BUILT-IN-TEST (BIT) | | 5 TYPE OF REPORT & PERIOD COVERED<br>In-House Report<br>6. PERFORMING ORG. REPORT NUMBER<br>N/A |
| 7. AUTHOR(s)<br>Anthony Coppola | | 8 CONTRACT OR GRANT NUMBER(s)<br>N/A |
| 9 PERFORMING ORGANIZATION NAME AND ADDRESS<br>Rome Air Development Center (RBRT)<br>Griffiss AFB NY 13441 | | 10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>23380212 |
| 11 CONTROLLING OFFICE NAME AND ADDRESS<br>Rome Air Development Center (RBRT)<br>Griffiss AFB NY 13441 | | 12 REPORT DATE<br>April 1979<br>13 NUMBER OF PAGES<br>120 |
| 14 MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*<br>Same | | 15. SECURITY CLASS. *(of this report)*<br>UNCLASSIFIED<br>15a DECLASSIFICATION/DOWNGRADING SCHEDULE<br>N/A |

16 DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

Same

18. SUPPLEMENTARY NOTES

None

19 KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

| | |
|---|---|
| Built-In-Test | Fault Isolation |
| Testability | Performance Monitoring |
| Maintainability | Design |
| Fault Detection | |

20 ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report summarizes available information of use in designing built-in-test (BIT) capabilities in electronic systems. It describes the various types of BIT, design considerations and examples, data used in BIT design, display options, coupling and shielding considerations, and optimization models.

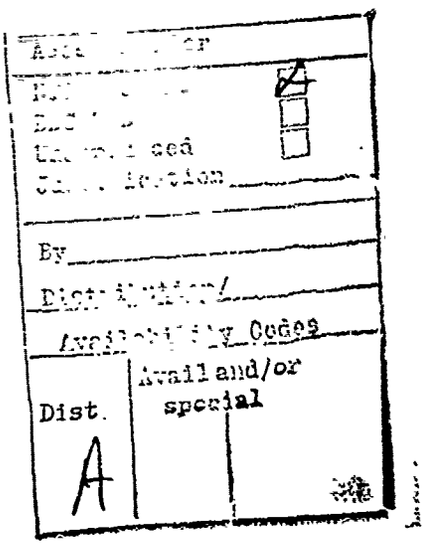
DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

## TABLE OF CONTENTS





**FIGURES** **PAGE**

**TABLES** **PAGE**

# SECTION I
# INTRODUCTION

The design of BIT is a complicated process. BIT (Built-in-Test) includes every concept used to detect and isolate a fault without the use of external test equipment, and it ranges in complexity from the indicator lamp that lights when an equipment power switch is turned on to the use of a resident computer for the generation of test signals and evaluation of system responses. BIT can be continuously operating, interleaved with other operations or initiated on command. It includes hardware sensors and software error correcting codes. Its particular mechanization and utilization in a system are, of course, determined by the designer and careless design of BIT can be a needlessly expensive experience. The purpose of this manual is to provide some guidelines to the designer to help him recognize his options and select the optimum approach for his system. It is a distillation of available information, drawing principally on NAVMATINST 3960.9, Built-In Test (BIT) Design Guide, 9 September 76, by the Naval Electronics Laboratory Center; a Study of a Standard BIT Circuit, February 77, by Research Triangle Institute for the Naval Avionics Facility; and RADC-TR-71-281, Design of Integral Sensor Test Equipment, December 71, by Westinghouse for the Rome Air Development Center. The distillation and organization were prepared by Rome Air Development Center in response to a request by the Electronic Systems Division (ESD).

# SECTION 2
# ORGANIZATION

The following sections of the manual will provide in sequence:

| SECTION | TOPIC |
|---|---|
| 3 | A Discussion of the Various Types of BIT and Their Application. |
| 4 | General Design Considerations. |
| 5 | BIT Design Examples. |
| 6 | Data used in BIT Design. |
| 7 | Display and Evaluation of BIT. |
| 8 | Coupling and Shielding Considerations. |
| 9 | Analyzing and Optimizing BIT. |
| 10 | References and Bibliography. |

# SECTION 3
# CATEGORIZATION OF BIT

BIT can be categorized in several different ways, such as:

a. Functional levels tested.
b. Purpose (detection, isolation, correction, prediction).
c. Active vs passive.
d. On-line vs off-line, vs interleaved.
e. Inductive vs deductive.
f. Centralized vs decentralized.
g. Hardware vs software.

This results in a multi-dimensional matrix of possible BIT classes. Each class has its use and the designer's job is to determine which class or, in most cases, which combination of BIT classes will fill his needs in the most cost effective fashion.

To illustrate various considerations in selecting BIT, let us start with an illustrative BIT system and then consider possible alternatives which could have been applied.

Figure 3-1 represents a distributed BIT system. Each module contains its own BIT circuitry and each module can be interrogated by a BIT microprocessor. Status signals are collected in OR gates and a system failure is indicated when one or more signals are present.

Let us now compare Figure 3-1 against the list of categories presented at the start of the section:

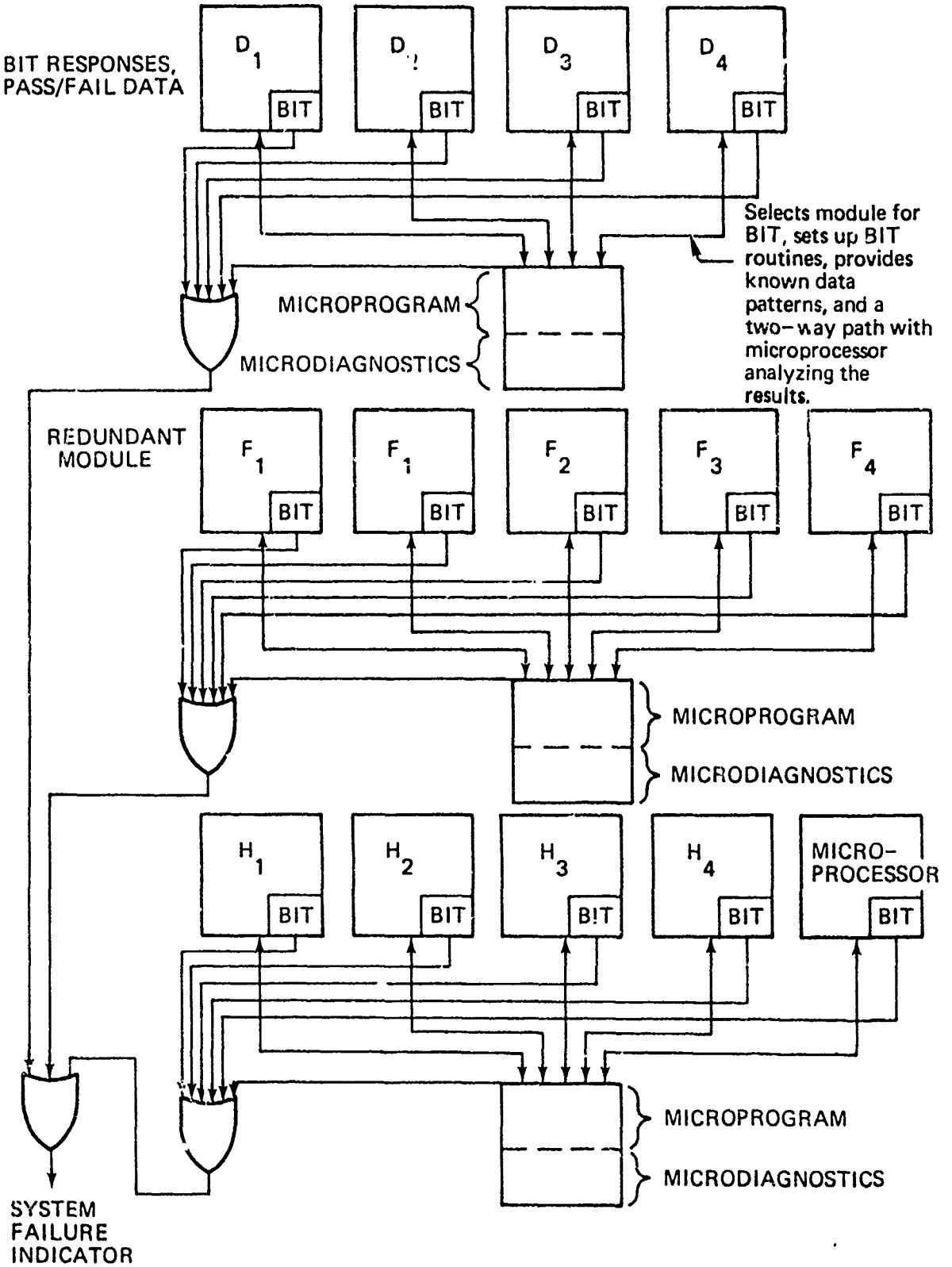



**FIGURE 3-1 DISTRUBUTED BIT SYSTEM**

**3.1 Functional Level Tested:** Figure 3-1 tests at the module level. BIT can be designed to test at the system level, subsystem (e.g., the four models $D_1$ $D_2$ $D_3$ $D_4$ might be a subsystem of Figure 3-1) module level, printed circuit board level, or even part level. The level depends on the reason BIT is employed. For example, monitoring the output of an engine generator is sufficient to determine whether or not it is working. However, if rapid restoration or prediction of impending failures is required, monitoring of subelements such as the oil system, fuel system, windings, bearings, etc., would be desirable. For the system in Figure 3-1, if this were an inaccessible system such as a satellite, module level tests would not be necessary (presuming it is feasible to assure system health without lower level testing). On the other hand, if it were a repairable system and maintenance is performed by replacement of subunits of each module, (e.g., printed circuit boards in $D_1$), it might be worthwhile to extend BIT to a lower level. This would depend on the maintenance time requirements and the life cycle cost trade-off between more complexity and easier maintenance. If the maintenance philosophy is to replace a defective module, the BIT design of Figure 3-1 would permit rapid isolation of the failed module and ease the on-line maintenance. Even then, however, lower BIT might be more economical by reducing maintenance costs at the next level of maintenance or perhaps permitting the elimination of a level of maintenance. Hence, maintenance requirements and life cycle cost trade-offs must be always considered in selecting the functional level tested. This will also be true of selections from the other categories.

**3.2 Purpose:** At least two purposes are served by the BIT system of Figure 3-1. It first detects a failure. The system failure indicator is triggered by a failure of any of the modules. In some cases, this might be all that is necessary (for example, to alert a pilot of a navigational computer failure). The system in Figure 3-1 also serves the function of isolation to a module level, and hence, aids maintenance. This, for $C^3$ systems where rapid restoration is critical, improves operational availability as well as reducing maintenance costs. Another function of BIT is correction. In Figure 3-1, $F_1$ is a redundant module. The BIT could be designed to correct a failure by triggering a transfer of use from the failed module to its redundant partner. Other BIT methods for correcting failures are error correcting parity codes and majority voting circuits. These also increase operating availability by "working around" a failure. Failure prediction can also be performed by BIT in some cases. Returning to the engine generator example, vibration monitors or oil analysis devices can signal the need for maintenance before a failure occurs.

**3.3 Active vs Passive:** In Figure 3-1, the interrogation of the modules by a microprocessor which then evaluates the response, is an example of active BIT. Passive BIT monitors system performance without the use of a test pattern generator. An example of active BIT might then be reading a test pattern into a memory and then comparing a readout to the test pattern. Passive BIT might be exemplified by a parity check evaluation. The attribute that distinguishes active BIT from passive BIT is, in fact, simply a test generator. It is a nontrivial task to determine a meaningful test set (i.e., a set of input test vectors) for each module in a system. However, once that test set has been determined, it is generally not particularly difficult to develop hardware to present the test set as input conditions. Figure 3-1 shows both active and passive BIT evaluating the same modules. This is not a duplication because the passive BIT may not be able to completely monitor the module, while the active BIT cannot be used without disrupting the modules operation. Hence, the passive BIT provides continuous, but incomplete evaluation, while the active BIT compliments it with more complete, but not continuous evaluation, which leads directly into the next category.

**3.4 On-Line vs Off-Line vs Interleaved:** In Figure 3-1 the passive BIT is on-line, that is in operation while the system is operating. The active BIT is off-line and checks the modules when the system is not performing its mission. It is also possible that the active BIT is used while the system is operating, but the module is not. Not all modules are operated continuously and a computer controlled BIT system can take advantage of times when a module is not needed to run a test sequence. It is also possible that the whole system can be tested without interrupting operation (e.g., using RADAR "dead" time). The latter is referred to as interleaving BIT. Interleaving can be a powerful means for maintaining confidence in a system without

disrupting its mission for running tests. Another type of BIT is initiated BIT which is used only when called upon manually or by a computer response to an indicated failure. The designer must use good judgement in arriving at the best combination of BIT. One popular approach is to use continuous on-line BIT to monitor the general well-being of the hardware, and to use initiated BIT to assist in locating the malfunction. Initiated BIT is also very useful in testing sections of the unit which, if tested continuously or periodically, could disrupt the normal flow of operation. Initiated BIT can be handled by a computer.

**3.5 Inductive and Deductive BIT:** Deductive BIT philosophy assumes that if *a certain function* is within its stated tolerance limits, then *all the variables* involved in generating that function must also be within their stated tolerance limits. If, for example, a radar klystron transmitter output is measured and found to be normal, it is assumed that the intervening waveguide elements are not excessively lossy and that the klystron power supply is functioning satisfactorily.

Inductive BIT philosophy concludes that if a *a specified set of measured functions* are found to be within their stated tolerance limits, then a *single unmeasured* (and perhaps unmeasurable) *function* also must be within its stated tolerance limits. Thus, if measurements taken at a waveguide bi-directional coupler show that the microwave power and frequency are correct and that the standing-wave ratio is not excessive, it is concluded that the antenna radiation is in accordance with specifications.

In Figure 3-1, system performance is apparently judged by induction. If no module sends a failure signal the system is considered operational. However, the BIT in each module can be either deductive or inductive. A parity check used as BIT deduces from a successful test that the module is working properly. An active BIT check of a memory successfully reading out a previously entered test pattern induces that the memory will function properly with a mission memory pattern.

**3.6 Centralized vs Distributed:** The passive BIT in Figure 3-1 is distributed. Each BIT circuit has as its only function the test of the one module in which it is contained. The active BIT is also distributed in the system, but centralized in each subsystem, with a microprogram and microdiagnostics for each of the three subsystems. However, if these functions were contained in one microprocessor, it would be a centralized system BIT (note that the distribution of BIT is not necessarily the same as the distribution of fault indicating signals. Distributed BIT can provide a centralized fault signal as shown in Figure 3-1 and centralized BIT can trigger distributed fault indications). Centralized BIT in a computer controlled system can have advantages in that the computer can better interleave tests with system operation and that less hardware is required. Distributed BIT has the advantage that a subsystem can be taken off-line and not require the use of the system computer for diagnosis of failure and verification of restoration. Combinations are possible with the central computer directing distributed BIT systems to operate and report findings. The decentralized control must have the following characteristics:

(a) Self-test capability.
(b) Isolation from other system data.
(c) Some type of synchronization with system functions.

**3.7 Software vs Hardware:** Figure 3-1 utilizes software BIT in its active bit circuitry. The BIT portion of each module can be software or hardware BIT, though it is most likely hardware. Software, of course, implies hardware in the form of a computer or microprocessor. If a computer can be shared between operational and test functions, the hardware costs associated with BIT become minimal. There may be, nonetheless, some requirements for sampling and buffering signals and for interfacing signals which are not normally accessible to the computer. Where existing system computer resources are not available, the microprocessor is an interesting candidate for BIT implementation.

Hardware in the form of hardwired logic using circuits which are standard in the system can also be used to realize BIT. With adequate operator procedures and troubleshooting aids, such as Fault-Location Charts, hardware BIT is adequate for some applications.

Software BIT offers many advantages. Among these is the ability to make changes by reprogramming as the system is modified or as experience suggests. In addition, software BIT may be made more comprehensive using a fixed amount of hardware. Software BIT is particularly applicable to end-to-end testing. It can provide input stimuli to the system under test and can monitor the output. It can determine a GO/GO-NO condition and also provide diagnostics to isolate the fault to a functional area. However, the use of a computer does require the fabrication of special interfacing hardware.

Several considerations should be kept in mind when designing a system utililzing software BIT. These are:

(a) It is essential to isolate the system data from the test data. This is true for both the input as well as the output. If the test signals are not inhibited at the output, they may be misinterpreted by the interfacing hardware as a command or as fraudulent data;

(b) When monitoring the output of a given functional area, it is essential to provide adequate tolerance. The amount of tolerance will depend upon the specific application. An important point is that BIT tolerance should never be less than that of test equipment used in off-line maintenance, or it will result in indicated failures not verifiable later on;

(c) The input stimuli should be kept at a minimum level to minimize their effect upon performance. Input stimuli should be chosen to closely resemble normally accepted data. Therefore, if data are supplied inadvertently to an interfacing unit, a malfunction will not be created in that unit. Also, stimuli should not be selected so as to cause fraudulent commands which might be detrimental (i.e., firing commands, etc.);

(d) Existing data networks should be used wherever possible to reduce cost and provide testing of interface circuitry;

(e) The key to optimized fault isolation is judicious selection of monitoring points. Wherever possible, a common monitoring point should be utilized to test more than one functional area; and,

(f) The possible increase in computer size which may be necessary for the inclusion of BIT must be considered when initially designing the total system.

Hardware BIT is especially useful in functional areas which perform signal format transformation (e.g., analog-to-digital conversion, digital-to-analog conversion, logic level to power controller). Hardware has its greatest value in areas where software cannot be used efficiently. In these areas, the hardware BIT can monitor the input and output and ascertain whether the output result is correct. To reduce circuit complexity, it is often advantageous to have the hardware BIT monitor several functional areas simultaneously on a time-sharing basis. This of course implies a sequencing program (software).

# SECTION 4
# GENERAL BIT DESIGN CONSIDERATIONS

## 4.1 Design Philosophy:

BIT cannot be considered an afterthought; it must be an integral part of the design. Overall performance tests must be carefully planned to ensure that the machine is properly exercised and that its essential functions are properly observed. The test conditions must truly reflect operating conditions. The consequences of an incomplete BIT may prove disastrous.

Fault localization requires access to points within the machine which would not be required under normal operation. Here, we must contend with the uncertainty principle—what we wish to observe we may also affect. A common example of this is the capacitance of a probe which may introduce enough delay to cause the test circuit to malfunction Test points for BIT must be included in the original design.

BIT often means additional hardware above and beyond that required for the primary function. Reliability and cost are affected and trade-offs leading to a balanced solution must be made.

BIT protective circuitry should be designed to be fail-safe. This means that, in the event of wire breakage or a connector being left disconnected, the fault will be detected and the system will be protected. In addition to BIT circuitry which actuates visual status indicators, BIT features may also include test points and self-test meters. It should be remembered the goal of BIT design is to decrease the mean-time-to-repair (MTTR) by steering a technician to the faulty component as quickly as possible. Knowing this, the good BIT designer will attempt to attain his goal through various means with innovative circuitry and by rearranging circuits to perform dual functions with a single circuit if possible, such as driving a visual indicator and tying into various AND gates all with a single driver. He will also standardize his BIT circuitry as much as possible. This drives down the cost of implementing BIT and falls right in line with an important goal of any designer or businessman—which is to try to give the customer the most for his money.

Other important general considerations in designing hardware BIT are:

(a) The reliability of the BIT should exceed that of the hardware being tested. If this is not the case, the probability of failure of the BIT may be almost as great as the probability of failure of the unit under test;

(b) The BIT should be kept simple but also effective as required to meet operational needs;

(c) Failure in the BIT circuit should not affect performance of the system. Whenever feasible, the BIT input and output should be sufficiently isolated from the normal channels so that any failure in the BIT will not cause the function under test to be impaired;

(d) The type of circuitry used for BIT should be, if feasible, of the same type used in the normal system to minimize the number of different types of components used in any particular system; and

(e) The use of high-reliability and burned-in parts as well as integrated circuits (ICs) is highly recommended.

## 4.2 False-Alarm Minimization:

All possible input-stimuli combinations, when economical, should be considered to eliminate the

possibility of a good condition fraudulently indicating a fault. This can happen easily when the monitoring circuitry examines only certain combinations of input stimuli. When environmental conditions are proper and logic design is performed correctly, false-alarm signals typically arise from three causes: a logic element has failed; sufficient noise has entered a circuit; or the logic is improperly cooled. Self test of BIT is an important consideration since an undetected BIT failure which incorrectly indicates a system failure will increase maintenance time by directing the maintenance man to the wrong action. Good design practices apply to BIT design as well as system design and that includes unambiguous signal interpretation in the BIT design.

### 4.3 BIT Performance Monitoring:

It is desirable that all sub-assemblies within a unit be tested with some level of BIT. BIT has two primary functions:

(a) To monitor the "general well-being" of the hardware and inform the operator of any malfunction: and

(b) To aid in the location of failed components.

The "general well-being" of a system or subsystem is monitored by means of end-to-end BIT. Here stimuli are applied to the front end and the responses are measured at the output. The creditability of this kind of test will depend upon how much and in what manner the system hardware is exercised and the amount of system complexity.

In addition to the overall system end-to-end tests, there may be similar tests for the subsystems and lower-level modules. For detection of failure, the subsystems could be sequenced automatically (or by operator selection) through their tests to provide the first step in fault localizations.

Continuous or on-line performance monitoring may be achieved by: (a) observing normal input and output signals and applying reasonable criteria based upon known system-transfer characteristics, and (b) time-sharing the input with test signals and examine the test response at the output. This is possible if the system can accommodate, on a time-shared basis, both the normal and test inputs.

With judicious selection of input stimuli and careful monitoring of the mode of failure of systems, it is usually possible to locate the malfunction to a functional block. This functional block could be as simple as a single curcuit, or as complex as a group of several cards.

All prime functions of any system should be tested from end-to-end to verify that no malfunction will cause harm to personnel, damage to associated equipment, or jeopardize the completion of the mission. For these reasons, it is important that primary functions, such as fire control, be built to incorporate fail-safe operation Also, the greatest priority should be placed on these systems to obtain the highest technical level of BIT which is practical. The ability to correct malfunctions need not be limited to replacement of failed parts, but may also include the ability to transfer to a redundant unit or channel. It may be desirable to incorporate redundant channels and automatic transfer of these functions, if cost and size permit.

The size and complexity of BIT will generally reflect the size and complexity of the system it is intended to cover. Reliability is generally accepted as being inversely proportional to system complexity.

## 4.4 Bit Fault Isolation

The level if BIT required to locate the malfunction is directly related to the MTTR requirement for the system. The total repair time is the sum of the following steps: location, isolation, disassembly, replacement, reassembly, and checkout. The time taken by the last four steps (before checkout) is independent of the level of BIT which is implemented. However, if these steps consume a considerable portion of the MTTR, the BIT level must be sufficiently high to locate the malfunction to the smallest functional block feasible. This functional block may consist of one of several cards. If this is accomplished, further isolation could be accomplished with the aid of test equipment such as oscilloscopes and voltmeters. It is not recommended that use be made of sequential replacement approach to isolate problems (replacing cards within a functional area until the malfunction is eliminated), since connectors have a comparatively high failure rate and a good unit can be faulted due to a malfunction elsewhere.

## 4.5 Parameters To Be Monitored

BIT, depending upon the type of equipment it checks, tests parameters which are key elements of the hardware. The number of tests required to detect all faults in large arrays using large numbers of integrated circuits frequently cause test times to be impractical. Shortcuts are commonly used to avoid long test times. Equipments and types of circuits are categorized as:

(a) Digital circuits, combinational and sequential,
(b) Analog circuits,
(c) Electronic equipments,
(d) Electromechanical equipments, and
(e) Non-electronic systems.

Electronic circuit test parameters include:

(a) Voltage sources,
(b) Current sources,
(c) Equivalent current vectors,
(d) Standard deviations of node voltages,
(e) Time,
(f) Theveinin equivalents,
(g) Frequency values,
(h) Nodal conductances, impedances, currents, and voltages,
(i) Branch voltages and currents,
(j) Word size,
(k) Element currents, voltage, and power losses,
(l) Nodal voltage sensitivities,
(m) Resistances,
(n) Betas, and
(o) Mutual inductances.

Digital parameters can be monitored as:

(a) Open—stuck at "1"
(b) Short—stuck at "0"
(c) Hole—closed
(d) No hole—missing work

Analog-circuit test parameters consist of the same parameters as those of digital circuits plus volume flow, pressure (hydraulic), torque, angular velocity, displacement (rotational), force, velocity (translational), temperature, stress, and strain.

Analog parameters can be monitored as:

(a) Open,
(b) Voltage
(c) Time,
(d) Ratio,
(e) Tolerance,
(f) Frequency,
(g) Power. and
(h) set level.

## 4.6 Bit And Redundancy

Standard techniques to maintain reliability in the face of increasing complexity include the use of redundant and alternate circuits and the incorporation of self-test capabilities. These techniques, until fairly recently, found limited acceptance because of the attendant rise in such factors as cost, weight, size, power consumption, and heat dissipation. Thanks to the rapid advances made in integrated-circuit technology, weight, size, and power consumption penalties have decreased.

The cost of BIT and the benefit of BIT in terms of reduced MTTR and increased availability must be traded-off against the possibility of redundancy to reduce system failures. By providing more than one functional path or operating element in areas deemed essential to system success, system reliability may be improved. BIT may utilize redundancy in a voting-circuit arrangement to combine BIT and redundancy for better system availability.

The use of redundancy is not a panacea for all reliability problems nor is it a substitute for good design. Redundancy increases system complexity, weight, size, and power consumption On the other hand, redundancy may be the only solution for a particular design. Trade-off studies of cost versus weight and size versus reliability should be conducted on redundancy versus BIT for a given system. Often times, a combination of redundancy and BIT may be the best solution. On limited-life items, such as SQUIBS, firing redundancy is employed since tests would not be practical. Finally, redundancy may do wonders for mission reliability but it increases both the maintenance workload and overall support cost.

## 4.7 Marginal Operation Detection

Proper BIT instrumentation will often provide for the detection of marginal situations which will alert an operator to an existing degraded situation, but will not require the system to be turned off. Examples of this might be in an antenna-drive circuit where a heater ceases to function properly in an airborne system, or in a transmitter where the output of an intermediate stage may be lower than specification, but due to extra gain in succeeding stages, the total output power is still above the minimum allowed to meet specifications. In these situations, an amber indicator (rather than a red indicator) will be illuminated to alert the operator to a degraded situation which soon could require attention. At the moment of alerting, it may be ignored until a more suitable time for examining or correcting the marginal situation comes into being. These marginal situation circuits are usually not tied, in the case of transmitters, into the transmitter GO circuitry. When the marginal situation amber fault indicator is energized and used in conjunction with the transmitter GO indicator, a more complete actual status of the overall assembly is presented.

## 4.8 Bit Commonality

Common hardware can often be put to a variety of uses. This is particularly true of computer-based systems. While programmed to fulfill different system requirements, many circuits can be exercised by means of standard BIT programs. The commonality of BIT hardware and software can be extended to all designs which used a standard computer and peripherals or which use a fixed set of functional modules.

## 4.9 Automatic Test Equipment Compatibility

In the consideration of BIT design, provision should be made for ATE accessibility both to initiate the BIT operation and to extract the BIT information. In addition, where BIT is used to detect a failure and ATE to isolate the failed part for repair, BIT threshold must not be wider than the ATE threshold. Otherwise, failures indicated by BIT will not be verified by the ATE and the unit returned to operation where BIT may again reject it.

## 4.10 Bit And External Testing

External automatic-test resources should be considered with proper interfaces incorporated in the BIT design so that BIT may include the capability of determining operational readiness of equipments as well as the localization of faulty components. Trade-offs should be made early in the system acquisition process to evaluate the effect of external testing upon operational effectiveness, logistics, maintenance, and system life-cycle costs. Usually, a combination of BIT and automatic-test-equipment (ATE) monitoring and diagnostics will be employed in a given system.

Incorporation of external automatic-test functions is costly and, if not properly integrated with the system, may induce more reliability and maintainability problems than it solves. However, external automatic-testing should reduce corrective maintenance time and increase system availability. Automatic-test features can be adapted to detect (or predict) impending failures. Corrective maintenance routines, wherein potentially faulty modules are replaced, will increase system availability. Automatic external fault-isolation techniques, augmenting BIT, can reduce both the number of maintenance personnel and the maintenance skill levels required to maintain the system.

## 4.11 The General-Purpose Computer As A Bit Component

The general purpose computer is an attractive candidate for BIT applications. Test algorithms or procedures can be programmed and easily modified as hardware is changed and as more efficient test techniques are developed. With a library of test programs, a wide range of equipment could be serviced by a single computer. Where a computer is an integral part of the system, it can be time-shared to perform BIT functions. Under some circumstances, a separate BIT computer may be justified. There are, however, some drawbacks to be considered;

(a) With BIT concentrated in the computer, no testing can be performed unless the computer is operating properly. This includes the computer main frame, the peripherals which interface the units under test, and the displays and controls which enable the operator to communicate with the system. It is first necessary to verify the proper operation of these components before BIT can be started.

(b) The computer has a particular set of interface characteristics, such as the I/O bus and the direct-memory access (DMA port), which are generally not compatible with units not intended to be linked to the computer. Special interface hardware has to be provided;

(c) The computer and the unit under test are not necessarily synchronous with each other. Separate clock frequencies or processing rates may be used and, if so, the computer cannot be expected to monitor performance on a clock-by-clock basis. It may be necessary to develop summary or status information for the computer and this requires additional hardware;

(d) The computer may be time-shared to perform a variety of tasks, and on-line monitoring would be occasional or sampled, rather than continuous; and

(e) The computer software expense has to be viewed warily. Software costs often surpass the acquisition cost of hardware. Depending upon program structure and detailed implementation, modification of BIT software may range from easy and inexpensive to extremely difficult and very costly. The general necessity for documenting program details in technical manuals or other technical literature often also represents an overlooked expense.

## 4.12 Digital-To-Analog Converters

The means of providing information to the repairman are varied. In cases where a general-purpose computer is used, the link is usually a keyboard control panel for inputs and various alpha-numerical displays such as printers, display panels, and cathode-ray tubes for outputs. These may not suffice. Local indicators and probe points should be considered. Also, information in digital form may not be readily assessed. It is difficult to determine whether a counting chain is working properly unless one looks at each stage of the counter and compares waveforms. If the counter stages are applied to the Digital-to-Analog Converter (DAC) and viewed on appropriate test equipment or counted digitally, the operation is immediately apparent. With the price and size of DACs continuing to decrease, one can liberally sprinkle them about and route the single-line signals from each to a central monitoring area. DACs would prove effective for both data and control-path analysis.

## 4.13 ROMS & PROMS

ROMs are another resource which can be applied effectively to BIT, particularly in the control area. If the behavior of a circuit under given input conditions is known beforehand, then it is possible to state the response at whatever level of detail is desired. ROMs can store the responses for test or perhaps limited operating conditions and the ROM output signals can be matched to those of the circuit under surveillance. Any discrepancy would be flagged as an error. The ROMs are sufficiently fast so that errors within a 100ns period could be detected. In addition to detecting errors, the ROM can also include information concerning the source of the error and, indeed, the identification of the defective module. ROMs can also be used to generate complex test signal patterns.

Field-programmable read-only memories (PROMs) have grown in use to the point where they have overtaken read-only memories (ROMs) in BIT systems. PROMs can be readily divided into two major classes: bipolar and metal-oxide semiconductor (MOS). The bipolar units offer access times of 40 to 100ns (20ns for ECL) and store 256 to 4096 bits. MOS devices have speeds of 230 to450 ns, hold 1024 to 8192 bits, and are usually erasable.

The erasable MOS units have a transparent lid and can be returned to their unprogrammed state by subjecting them to strong ultraviolet light. Unlike bipolar PROMs, they permit the manufacturer to program all of the bits, to test the alternating-current performance, and then to erase the unit before shipment. It is for this reason that the user can have confidence that the devices are fully programmable. At this time, there are no erasable bipolar units (EAROMs). Reliability tests of the erasable MOS units show failure rates of less than 0.003 percent per 1000 hours.

The major applications of PROMs are in minicomputers and microcomputers for resident members of a BIT system. They are used for program storage and in microprogramming. Bipolar units are used for minicomputer program storage. For MOS microprocessors, all of the devices have sufficient speed for the application. For microprogramming where the PROM access time enters directly into the Arithmetic Logic Unit (ALU) time, the emphasis is on speed so that users are asking for the fastest TTL-compatible units and are considering ECL PROMs for future applications.

## 4.14 Bit And Microprocessors

Large-scale integration has made the microprocessor readily available in the MIL-SPEC-versions and commercially available in packages designed for printed-circuit board mounting. The microprocessor, which consists of command and data registers, bit counters, an arithmetic logic unit, and a memory, is designed to function as a microcomputer when power, a clock signal, and input/output peripherals are supplied. Thus, it becomes possible to construct, on a single printed-circuit board, a computer which can apply a programmed test sequence while evaluating and presenting the test results.

An additional desirable feature of the microprocessor, when used in a BIT system, is its ability to be easily reprogrammed. This simplifies modification of the test sequence for continued compatibility with a multiple application system under changing mission requirements, or as design changes are affected. The flexibility afforded by programming also makes it easier to alter the test sequence for improved efficiency or effectiveness as may be required with time and experience.

The microprocessor is inherently multi-functional by nature and can be applied to solve many different problems. It can be used as a building block to create new functions by the programming (firmware) of its ROM. Its applications range from an intelligent terminal to a peripheral controller to a central processor. It is, truly, a universal building block for many sophisticated systems. The same basic hardware set can, through programming, be tailored to a specific task. Parameters of interest in microprocessors are: number of chips to perform CPU functions, pins/package, supply voltages, clock frequency, instruction-word size, main-program storage, data-word size, number of registers, interruptibility, basic instruction time, number of instructions, and software aids.

The microprocessors lends itself to large-scale integration. An LSI hardware set can be universally applied to reduce both design time and design costs. It will also provide a more uniform approach to BIT. Along with firmware which is specific to a given task, there could be a universal set of microroutines to exercise all components and localize errors to the smallest replaceable part. It is highly desirable to have the microprocessor callable from a host computer as well as local circuitry.

## 4.15 Monitoring Of Stress Levels

In transmitters, it is invariably a necessity to monitor stress levels on such components as high voltage capacitors, pulse-forming networks, transformers, and high-power high-cost amplifier tubes. The cost of a component and the ease with which it will be destroyed by the application of improper stress levels usually are the prime determinants as to whether or not a stress level shall be monitored. Most passive components are usually monitored for over-voltage, over-current or over-temperature stress only.

Limits on the percentage of voltage overstress allowed on a component are determined somewhat by the transient levels expected to be encountered and through which an assembly must operate without serious degradation in performance. Typically, in transmitters, logic supply voltages may be protected for 10-percent over-voltage or over-current.

Active high-voltage components, especially some high-power traveling wave tubes (TWTs), must be operated with a relatively narrow window of voltages and currents. Amplifier TWTs are subject to self-destruction if they are operated at a somewhat lower than normal voltage and oscillations start. The microwave circuit structure inside the tube usually overheats. Conversely, when a TWT is operated at a voltage only slightly above normal, usually only reduced gain results. But for operation much above normal, the tube may exhibit a tendency toward internal arcing. It is not uncommon for TWT cathode voltage monitoring circuits to require a precision of approximately plus or minus one percent.

For the coarse upper-voltage monitoring circuit in transmitters, the trip point typically is set for an over-voltage trip point of 10 to 20 percent above normal. Exceeding this level normally indicates one of two things. Either the high-voltage regulator circuit is defective, or a line transient is present with which the regulator circuit cannot cope. In either case, the sensing of this fault is extremely important and usually calls for immediate action in the form of a crowbar of the high-voltage power supply.

The most critical of all measurements in a transmitter is a current monitoring of the final tube cathode current. When pulse droop or ripple voltage on a transmitter must be exceptionally low, power-supply capacitors become large. The total energy stored in the energy-storage capacitor becomes large compared to the tube's discharge-dissipation ability. When this situation exists, a precision peak-current monitoring circuit, which will react quickly enough to divert excessive current from the tube and reliably switch the stored energy through a triggerable spark-gap within several milliseconds, becomes desirable. This circuit is the one which demands the swiftest response in a transmitter and, along with other crowbar signals, requires faster response than a computer controlled BIT with its relatively slow response time can provide. Here is another place where the special BIT circuitry of the fault processor performs a vital function.

## 4.16 Transducers And Sensors

In many cases, for BIT to function effectively, dependence must be placed upon sensors to detect electrical signals. These electrical signals can come from various tranducers such as diodes, thermocouples, encoders, synchros, resolvers, rotary variable-differential transformers, read-only memories, and dozens of other sensor types.

Items to be considered in the selection of transducers are:

(a) Sensitivity, which is the ratio of output to unit input and which must be adequate for system resolution requirements:

(b) Range, which must be capable of encompassing the maximum values to be measured. Maximum signal range is specified for a given noise level and the ability to discriminate low-level signals in the presence of a high-level signal is a requirement;

(c) Physical properties, which must be compatible with the working environment (melting point, freeze point, dissolve point, etc.);

(d) Loading effects and distortion must be minimal so that the transducer will have negligible effect upon measured or monitored sources of the phenomenon under study. Alternatively, the degree and nature of transducer influence may be sufficiently well known that calibration techniques can be used to give accurate data;

(e) The sensing and transduction elements must have sufficient frequency response in order to respond accurately in combination at the maximum rate of change of the effect under study;

(f) The electrical output format must be compatible with the measuring system in terms of detection and/or sensitivities;

(g) The output impedance must be compatible with succeeding electrical stages of the measuring system;

(h) Reliability and maintainability factors must not contribute to unreliability or unavailability within the using system;

(i) Power supplies for the transducers must meet the requirements for voltage, current, impedance. regulation, ripple, and transients;

(j) Sensing and transduction elements must be selected, designed, and shielded to reduce noise (any undesired signal) to negligible levels;

(k) The degree of uncertainty in transducer response (error or accuracy) must be compatible with the accuracy required for the measurement;

(l) The properties of some transducers drift with time and the frequency and desirability of recalibration must be determined; and,

(m) The transducer must be compatible with the environment in which it is expected to be employed. factors such as temperature, humidity, dirt, shock, vibration, electromagnetic fields, and radiation must be considered.

Specific types of sensors which may be used include the following:

(a) Linear Variable Differential Transformer (LVDT) Unit—an inductive transconduction element device requiring AC excitation. The frequency component is the linear motion to be measured, the frequency response is limited to about 50 Hz, response to accelerations from 0.1 to 500 g is possible, and the output voltage depends upon the type of excitation and associated circuits. It is capable of very fine resolution (microcentimeters) and its principal source of error is usually a stray magentic field.

(b) Potentiometric Transducer (motion)—the movable contact of a potentiometric or resistive transduction element coupled to a seismic mass. It is low in cost, typically free from undesired responses especially those of accelerations in other than the desired direction, its frequency response is limited to about 10 Hz, and it responds to accelerations from 0.1 to more than 100 g. Provision must be made to protect the mechanical linkage from accelerations greater than the intended range of the instrument.

(c) Strain Gauge (acceleration)—a resistive transduction element coupled to a seismic mass but without the sliding contacts of potentiometric devices. Its frequency response is limited to about 350 Hz, it responds to accelerations from less than 0.1 to more than 100 g, it produces output voltages in the range between 2 and 600 mV depending upon excitation (with bonded gauges giving lower outputs), it is linear over a wide input range, its excitation frequency must be at least 10 times the maximum rate of change of measured position, and it is sensitive to stray magnetic fields but can be shielded.

(d) Rotary Variable Differential Transformer (RVDT) Unit—similar to the LVDT except that the construction is designed for measurement of angular displacement.

(e) Synchro--a device having a three-phase stator winding and a single-phase rotor winding. When the stator is excited from a three-phase source, the rotor output is constant in amplitude, and its phase varies directly with position.

(f) Diodes—may be used as sensors since, in logic circuitry, they will provide voltage cutoff at required threshold thus enabling certain functions.

(g) Resolver—a device in which the rotor and stator have two-phase windings. The voltage developed by one rotor winding is proportional to the sine of the angular position of the rotor; the other rotor winding produces a voltage proportional to the consine of the rotor angular position. Any unused rotor winding must be short-circuited for proper operation.

(h) Induction potentiometer—similar to the resolver except that the output voltage is directly proportional to, instead of being a trigonometric function of, the rotor displacement.

Lately, particularly in airborne and shipboard systems, there has been great interest in monitoring stress, strain, temperature, vibration, and acoustic signatures of mechanical components such as gear boxes, transmissions, and engine fan blades. Sensors used in these BIT systems include strain gauges, thermocouples, acoustical microphones, and others listed in the references. These coupled with A to-D converters allow for signal comparisons and processing in the BIT system.

# SECTION 5
# BIT DESIGN EXAMPLES

This section presents some examples of BIT design, with varying degrees of detail, covering different applications of BIT.

**5.1 General Purpose Analog Sensors:** The sensor designs described below can be used individually or in combinations as "building blocks" to form a variety of sensors.

**5.1. Envelope Detector:** The circuit shown in Figure 5-1(a) is an active enveloped or video detector useful for RF signals from 5 to 50 MHz. By properly selecting ratios of $R^{1}$ and $R^{5}$, the gain can be selected anywhere between 0 and 26 dB. The envelope rise and fall times are less than one microsecond so the circuit is useful for detection of video pulses equal to or greater than one microsecond.

This sensor design can be scaled for operation below 5MHz by increasing the values of the capacitors in inverse proportion to the scaled frequency ratio. For example, if a video detector is required for a 500 KHz signal, all four capacitor values are increased by a factor of 10 (the ratio of 5 MHz to 500KHz). It should also be recognized that the detector rise and fall times increase proportionally so, in the above example, the circuit will detect video pulses of ten microseconds or greater.

The detector design utilizes a buffer amplifier (Q1) to prevent loading of the test point. This stage is an inverting amplifier whose gain, G, is

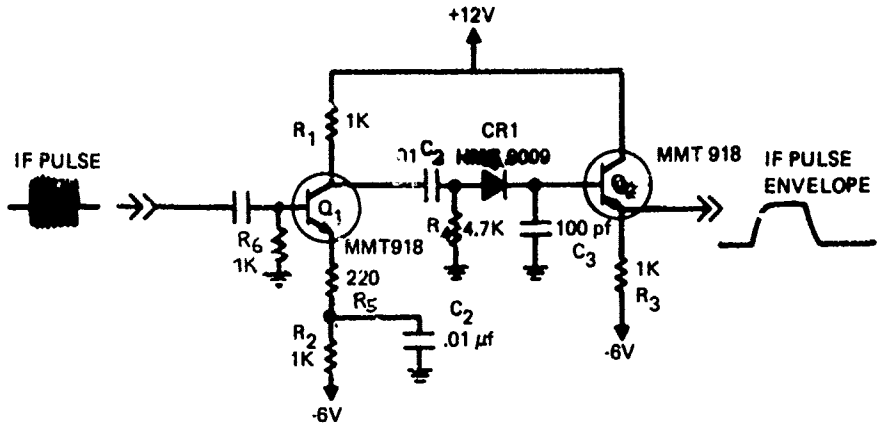
$$G = \frac{R_1}{R_2}$$

for all reasonable transistor betas. Sensor gain can be selected by varying $R_1$ while leaving $R_5$ at its nominal value. The collector of Q1 is AC-coupled to a conventional diode detector. The output state is an emitter follower to limit the loading of the detector and to provide a low impedance drive signal to the evaluator.

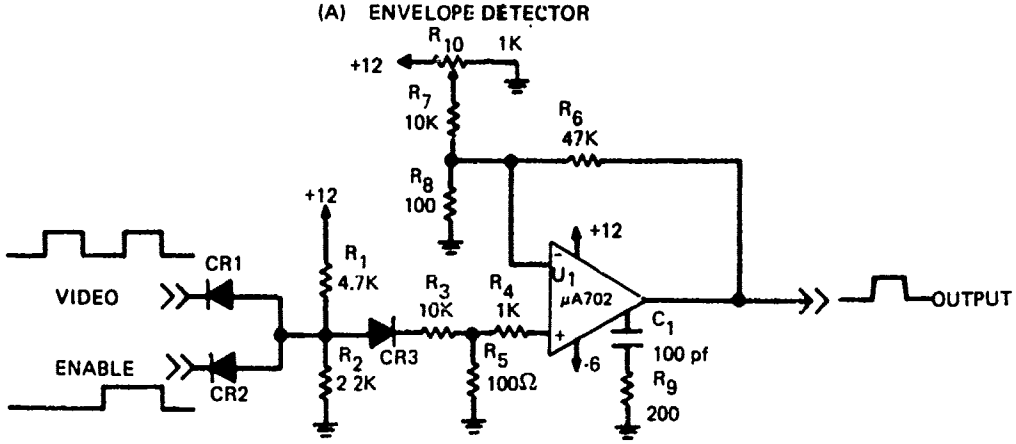
**5.1.2 Gated Video Buffer:** The circuit shown in Figure 5-1(b) is a gated video buffer for general use as a linear amplifier with an operational gating input. When the gating function is used, the amplifier output is zero except during an "ENABLE" signal when the sensor performs normally. To eliminate the gating function, one simply leaves the gate input terminal open and the circuit is continually enabled. The sensor is gated off by grounding the cathode of CR2 which shunts out the input signal and maintains the sensor output at zero. Sensor gain can be adjusted over a range of approximately 30 dB by altering the ratio of $R^{6}/R^{8}$. The gain equation is

$$G = 1 + \frac{R_6}{R_8}\,\frac{R_5}{R_3}$$

where $R_3$-$R_5$ form an input attenuator and $R_6$-$R_8$ provide amplification. The purpose of the input attenuator is to isolate the test point in the event of sensor failure. Adjustment $R_{10}$ provides a means for compensating the effects of bias current through $R_1$ and $CR_3$. The use of this bias circuit results in a linear amplifier even though the signals are injected through a diode gate.

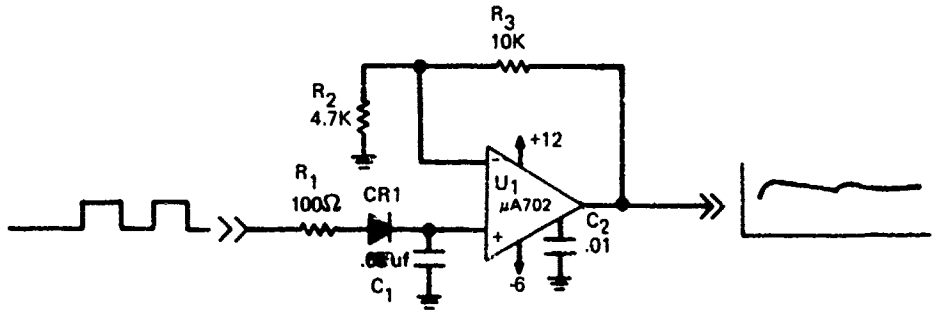



(A) ENVELOPE DETECTOR




NOTE: ALL DIODES HMG 9009

(B) GATED VIDEO BUFFER




(C) VIDEO DETECTOR

**FIGURE 5-1 GENERAL PURPOSE ANALOG SIGNAL AMPLITUDE SENSORS**

A compensation network $R_9$-$C_1$ is provided to make the 702 operational amplifier unconditionally stable. These values c e adjusted in specific applications by using the guidelines provided in application notes and specific .. heets for 702 operational amplifiers.

**5.1.3 ... Detector:** The circuit shown in Figure 5-1(c) is a general purpose video detector. The circuit c .. oe used as a peak detector for sinusoidal signals, square waves, pulse trains, etc. The detector discharge time is approximately 10 milliseconds so repetitive signals as low as one kilohertz will be detected and the DC output of the sensor will be directly proportional to the amplitude of the input signal. The upper frequency limitation is approximately one megahertz.

The circuit functions for input signals ranging from 1 to 4 volts. Circuit gain can be selected by adjusting the ratio of $R_2$ and $R_3$ in accordance with the following:

$$\text{Gain} = 1 + \frac{R_3}{R_2}$$

The circuit becomes a negative peak detector simply by reversing diode $CR_1$ but the gain equation is unchanged.

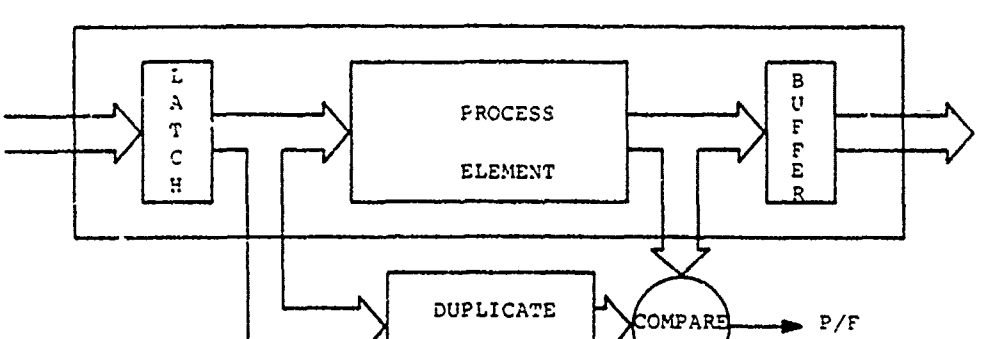
This circuit is compensated conservatively by $C_2$ for general purpose applications. The operational amplifier is simply a buffer for the already detected signal so there is no need for frequency response beyond approximately KHz.

**5.2 Replication:** One of the major approaches to mor. toring at any level is to replicate the whole or a part of an operational circuit and then compare the result of the two simultaneous computations. This technique may be done in many ways for any particular module. Figure 5-2 shows three examples of replication at various levels for a particular module. Each of these approaches has 100 percent input coverage, 100 percent cycle coverage, and varying degrees of function coverage with respect to the module level being monitored.

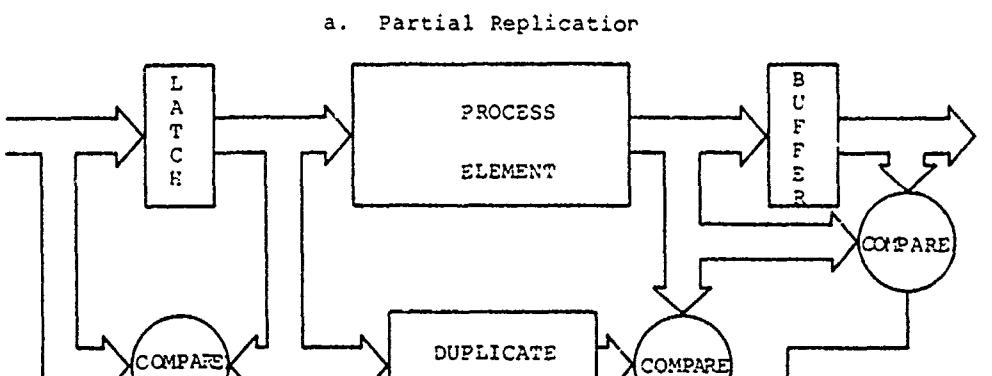
**5.3 BIT Coding:** Another approach quite similar to replication is the use of coding. A typical format for a coding based monitor is shown in Figure 5-3. The primary idea here is to find a code which can be compared in lieu of the full data result. The motivation for use of BIT coding is a monitor which may be nearly as effective but less costly than replicating and comparing as shown in Figure 5-2.

Monitoring performance through the use of error detecting codes introduces a subtle distinction not adequately characterized by the three ideas of input, function and cycle coverage. This can be understood by thinking of parity as a code checking device. While 100 percent of the input patterns may be monitored for 100 percent of the time, it is clear in the case of parity that one would be reluctant to say that there was 100 percent monitoring of the function fault behavior. The idea is that there are certain errors which will in fact disrupt normal results but will not appear in the coded form.

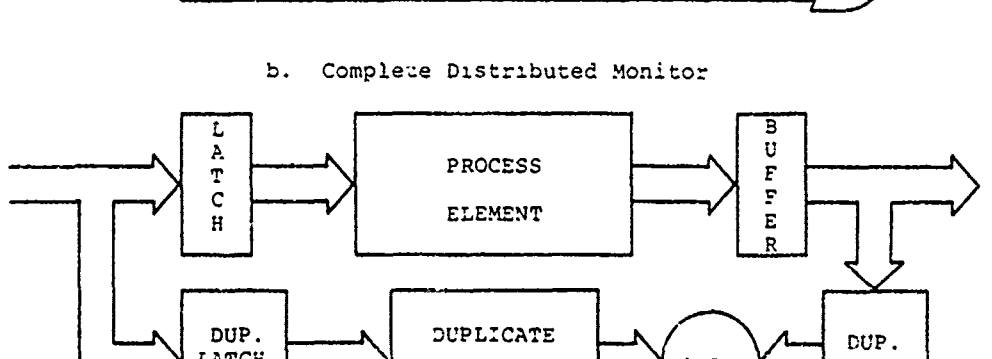
**5.4 Known Result:** The preceding two monitoring approaches are characterized by the fact that the desired answers for particular inputs are not known a priori. This places a computational demand upon the monitor to determine a correct answer to be used as a basis of comparison. It is possible to do monitoring of a simpler form when the desired answer can be established ahead of t' ne. Since this case is more restrictive, one might suspect the effectiveness is somewhat reduced. The motivation for pursuing this idea is based on the expectation that the cost of such a monitor would be significantly lower and its applicability more universal than the replication or coding approaches discussed previously.




a. Partial Replication




b. Complete Distributed Monitor




c. Duplicate Module and Compare

**FIGURE 5-2 TYPICAL BIT APPROACHES USING REPLICATION**

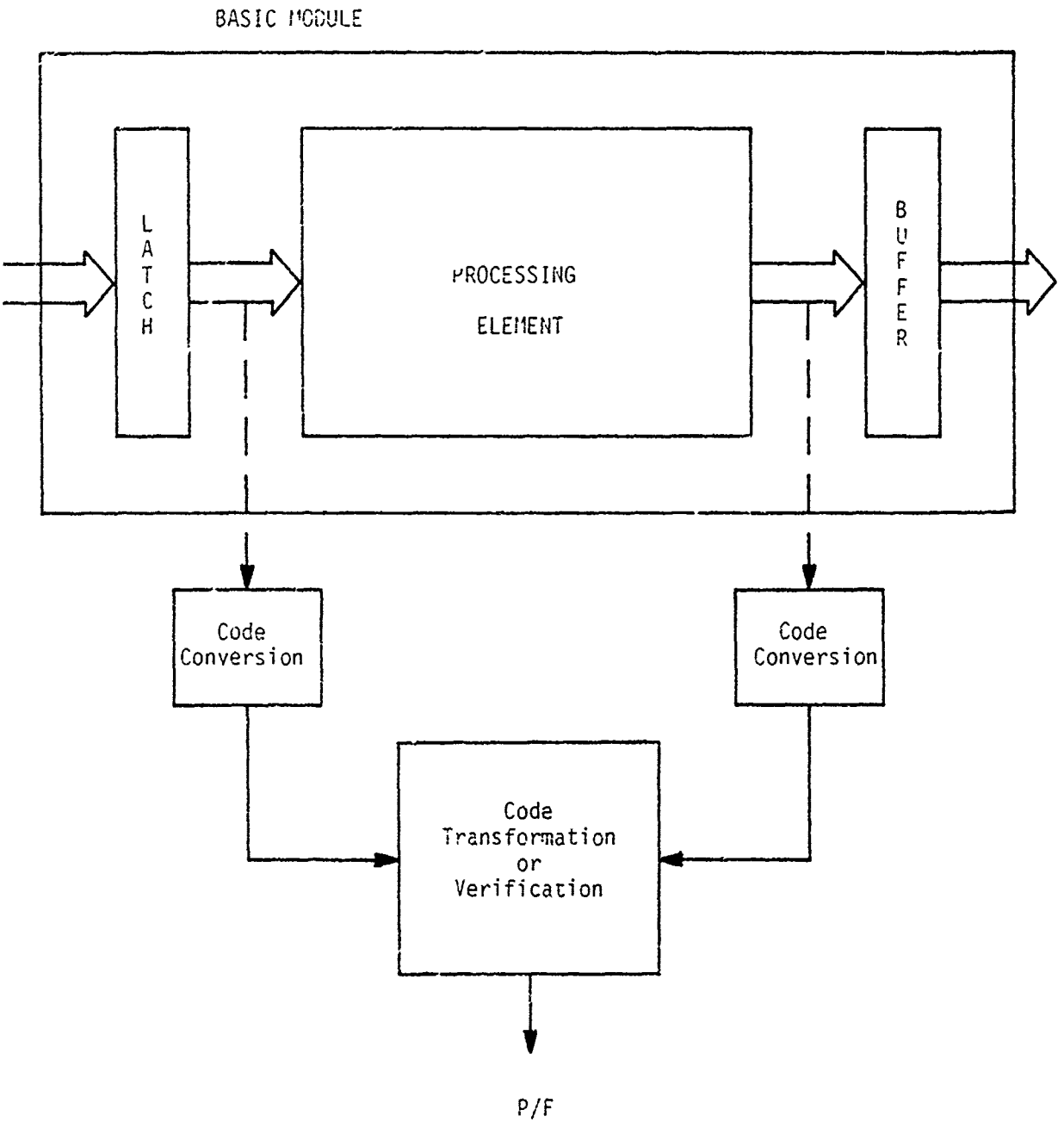



**FIGURE 5-3 VERIFICATION BY CODING**

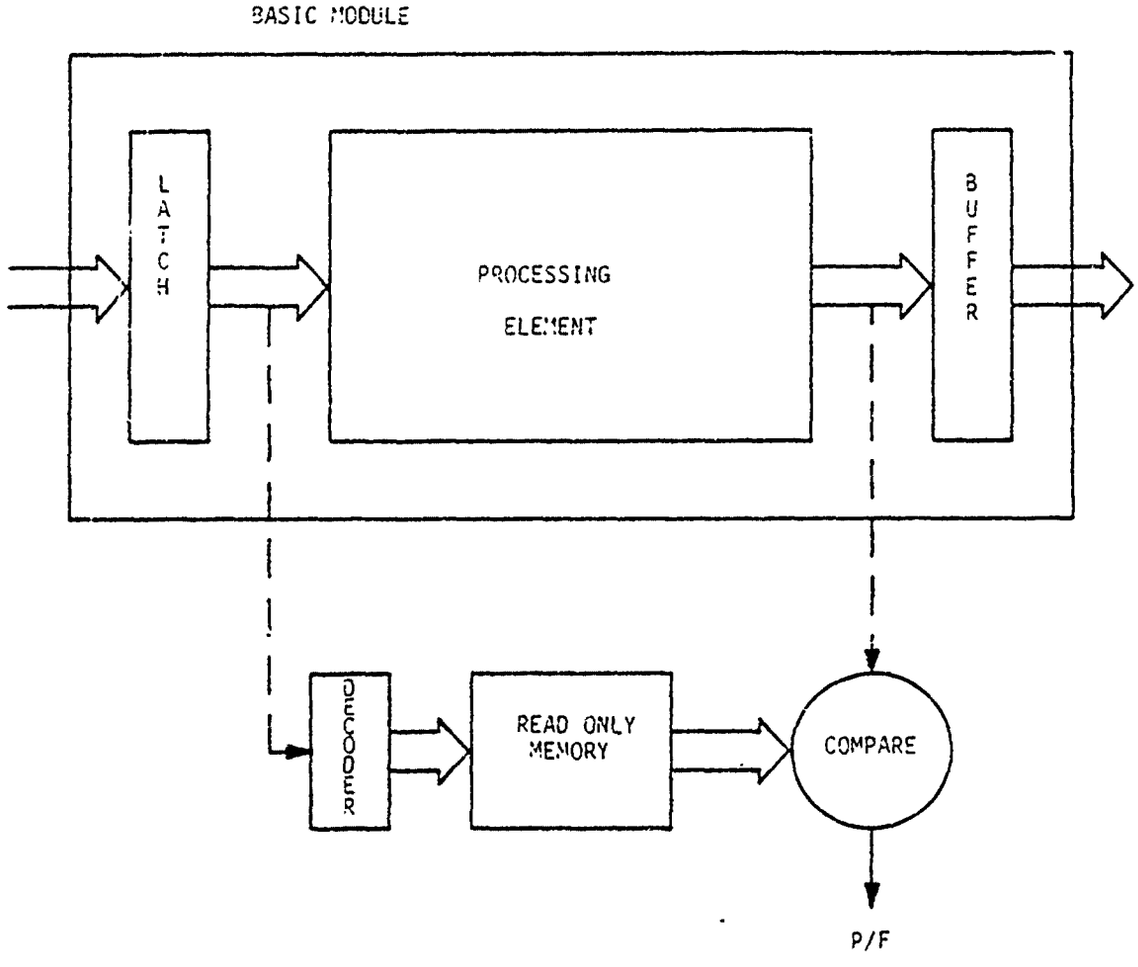



**FIGURE 5-4 KNOWN RESULT MONITOR WITH PARTIAL COVERAGE**

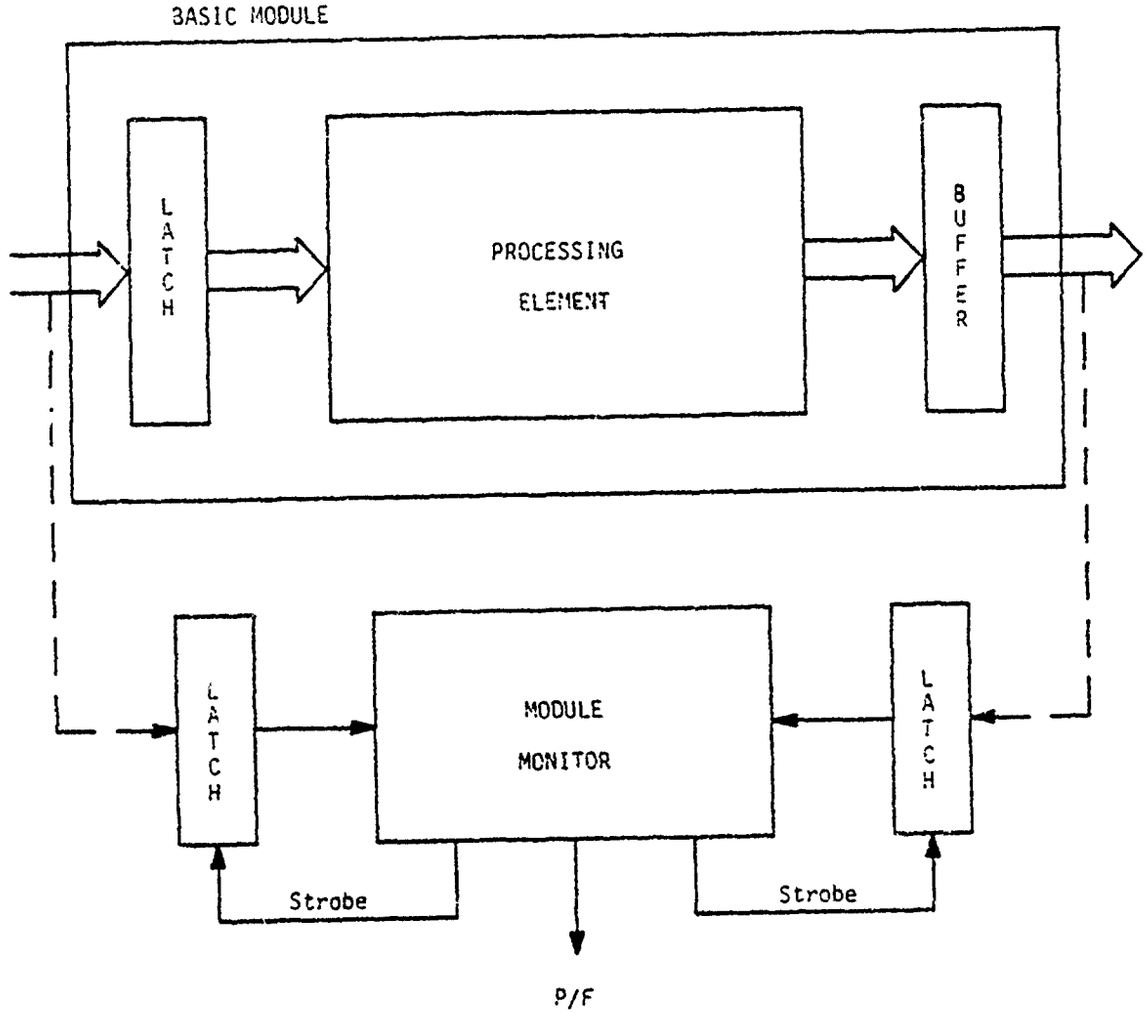



**FIGURE 5-5 SAMPLE MONITORING**

An example of a known result approach is shown in Figure 5-4. Since the total range of possible computations is large, there is motivation to provide fewer stored results than possible computed results. In this example the monitor checks for special cases of input occurrence, and, rather than compute a result to use as a comparison, a stored result is used. With a careful selection of input patterns, significant checking of the circuit performance can be accomplished.

As an aside, it should be noted that the effectiveness of this approach depends in large part upon the arrival of those special input cases at the module. In the case of a passive monitor, no control over the input set may be exerted. Therefore, the overall effectiveness of the known result monitor approach is a function of the data presented to the module under named system operation.

**5.5 Emulation:** To extend the idea of partial coverage, consider Figure 5.5 In this particular approach the cycle coverage is reduced from 100 percent. The basic operation of the monitor is to sample with particular attention given to the timing relationships between the input and the output. The sample monitor then computes, based on the input, its version of the output, and compares that to the sample output to determine the status of the network.

The primary motivation for considering this approach to passive BIT is that this particular task (that is. sampling and computation), is well suited to a programmatic passive monitor. A programmatic passive monitor could be utilized in a standard configuration to monitor a wide variety of processing elements or module functions.

Although an increase in flexibility of application is gained from use of programmatic devices, there will often be an attendant reduction in speed. Therefore, the concept of sampling becomes imperative since the programmatic monitor cannot, in general, perform computations in the same time frame as the operational modules in question.

The difficulty encountered in both the known result and the sampled monitoring approaches is that it becomes somewhat more difficult to define the effectiveness of this type of monitoring.

**5.6 Voting Circuits:** To eliminate the resultant BIT error caused by a failed logic element, voting circuits may be utilized. A 2/3 voting circuit is shown in Figure 5-6. This is more expensive, but it may be useful where a logic malfunction could result in severe damage or is imperative for the completion of a mission. In this case, the voting circuit consists of three identical components which function continuously and in parallell while a vote-taker circuit compares their outputs and selects the circuit output to reflect the majority opinion. The effects of a fault in any one component are eliminated.

A possible need for a BIT voting circuit might exist in the collector or anode coolant-flow sense circuit for a high-power microwave tube. If a circuit element failed, a steady coolant-flow GO signal would be indicated. This BIT circuit is normally not bypassed in a transmitter during actual battle conditions since almost immediate destruction of the tube will result. If, during battle conditions, a line was punctured and the coolant-flow was discontinued, the transmitter would not shut down and the final high-power tube would needlessly be destroyed within seconds.

Another battle situation example occurs if a shell fragment strikes and severs a wire from a single flow switch, in which case the transmitter would shut off needlessly. Depending upon the importance of this transmitter in the mission, a severe handicap would needlessly exist.

If three identical BIT coolant-flow monitoring circuits were installed with outputs each run into a 2/3 voting circuit, the failure in one coolant-flow sensing wire would enable the transmitter to continue to

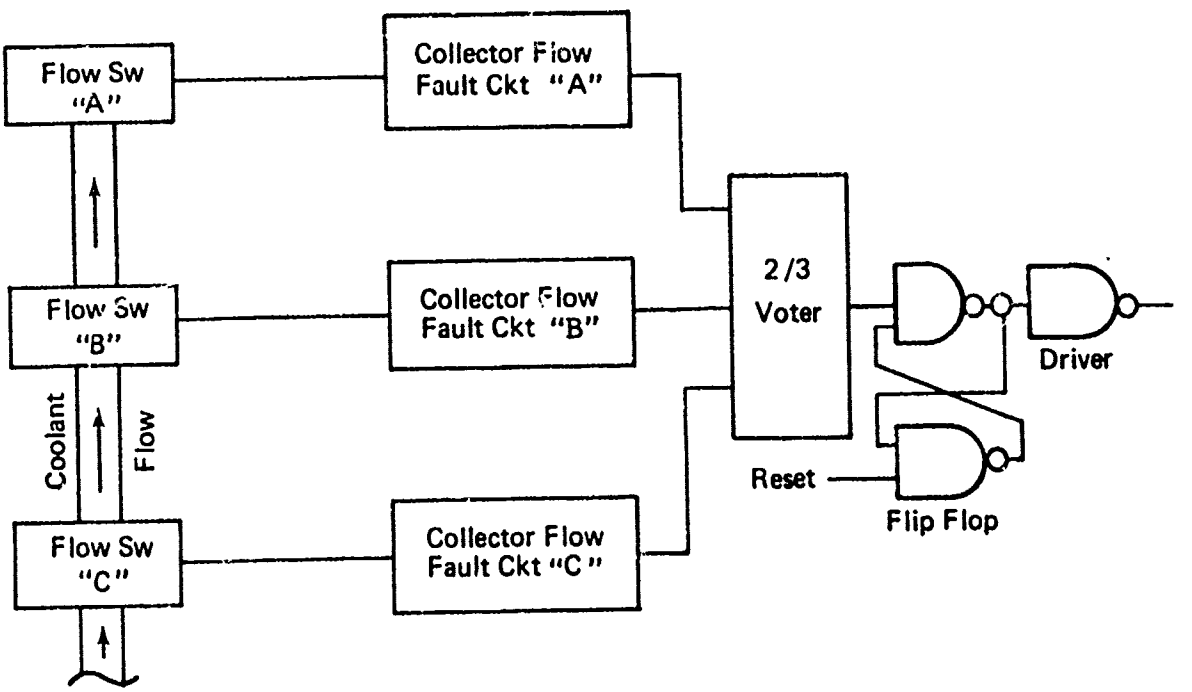



**FIGURE 5-6 VOTING CIRCUIT FOR THE MINIMIZATION OF BIT FALSE ALARM RATE.**

operate normally and the system would be able to complete its mission which could result in saving or reducing damage to a ship or aircraft. These voting circuits are not normally used in present designs, since they are not specified due to increased cost, space and weight, but they certainly offer significant savings in terms of readiness, logistics and maintenance cost.

Noise pickup in voting circuits from nearby circuitry such as power supplies or power lines can be reduced through the use of adequate magnetic shielding and filtering techniques.

**5.7 Built-In-Test for Memory Modules:** Memory modules will be examined in three groups according to function, as listed below:

**GROUP I**

**Random Access Memory (RAM) TTL**
**Random Access Memory MOS**

**GROUP II**

**Read-Only-Memory (ROM) TTL**
**Read-Only-Memory MOS**

**GROUP III**

**First-In-First-Out Memory (FIFO)**

The modules within each group provide the identical function to the user and therefore will be treated as one type of module. Any BIT technique that is beneficial to a TTL type memory will be equally beneficial for MOS memory monitoring.

The memory class modules are characterized by their ability to store data. This data can be program object code, numerical data, character representations, etc. The data storage format is in eight-bit bytes for all memory modules.

The BIT approach in this section for the memory modules includes a parity generator/checker as a standard approach to monitoring module interface circuitry and interconnections. This standard BIT circuit is described in Section 5.7.5. Additional BIT techniques which provide monitoring of data within the module are examined in the following subsections. This approach is based on Navy QED memory designs Variance from QED parameters may require modifications to the indicated BIT design.

**5.7.1 Random Access Memories:** There are two basically different approaches to checking memory modules that have contrasting effects on the system designer. One approach is off-line testing, which is when the module is tested, it is put into a test mode so that normal processing cannot be done. The

other approach is on-line. That is the module monitoring is performed on real data in real time with no resulting limitation on system throughput or overall speed. Of the various methods of on-line testing, replication and coding are the most practical approaches for RAM testing.

**5.7.1.1 Duplication:** Replication of the memory circuits is a straightforward technique for providing error detection. In the write mode each data word is stored in two separate memory circuits and when the data is read out of the memory module, the two are compared. Thus, any differences indicate a fault has occurred. This method of fault checking detects all errors in the memory including multiple bit faults and addressing errors. A block diagram of this expensive approach is shown in Figure 5-7.

**5.7.1.2 Parity:** Coding techniques offer a significant reduction in hardware over duplication at the expense of a decrease in fault detection capability. Single bit per word parity, the simplest coding technique, is also the most common memory error detection technique. To implement parity, an additional bit is added to each data word and the value of this bit is determined so as to make the sum of the number of one bits in the word an odd number for odd parity, an even number for even parity. When the data is read out, the parity is again generated for the data bits and compared to the stored parity bit. If these two parity bits are not the same, a fault is indicated. This coding detects all errors in an odd number of bits which include all single bit errors. A block diagram of a RAM module using parity is shown in Figure 5-8.

The QED memory modules are constructed using RAM circuits such that each memory integrated circuit contributes only one bit to the data word. It is therefore possible to detect both addressing errors and memory cell failures as long as there are an odd number of total errors in the data word. Single bit parity cannot detect faults in an even number of bits. It is possible to detect a greater number of multiple errors if a greater number of check bits are used. Unfortunately, these alternatives do not increase the fault detection capability as fast as hardware costs and will not be pursued at this time.

The question must be asked as to the likelihood of multiple bit failures in the memory modules currently being considered. The following provides some insight into the likelihood of multiple fault occurrence.

To compute the probability of occurrence of an even number of failures, consider the case of two simultaneous memory chip outages in a 1K × 9-bit RAM constructed from 1K × 1-bit memory integrated circuits. For a single data word to have two bits in error, the fault must occur at the same memory address. If $m_i$ represents the event that memory m has a failure at address i and if $n_i$ represents the event that memory n has a failure in location i, then the joint probability occurrence of simultaneous failure is

$$P(m_i \mid n_i) = P(m_i)\, P(n_i)$$

if the two events are statistically independent. If the probability of occurrence of a failure in any memory location is equally likely for each of two memory integrated circuits then the probability that the ith memory location will fail is

$$P(m_i) = P(n_i) = 1/N,$$

for an N- word memory. For example if N = 1024, then the probability of two simultaneous failures in memories m and n in the ith memory address is

$$P(m_i \mid n_i) = \left(\frac{1}{1024}\right) \cdot \left(\frac{1}{1024}\right) 1 \times 10^{-6}.$$

Since double bit errors can occur in any pair of memories, the combination of nine things taken two at a time, which is equal to 36, must be multiplied with the result of the above equation to obtain the probability

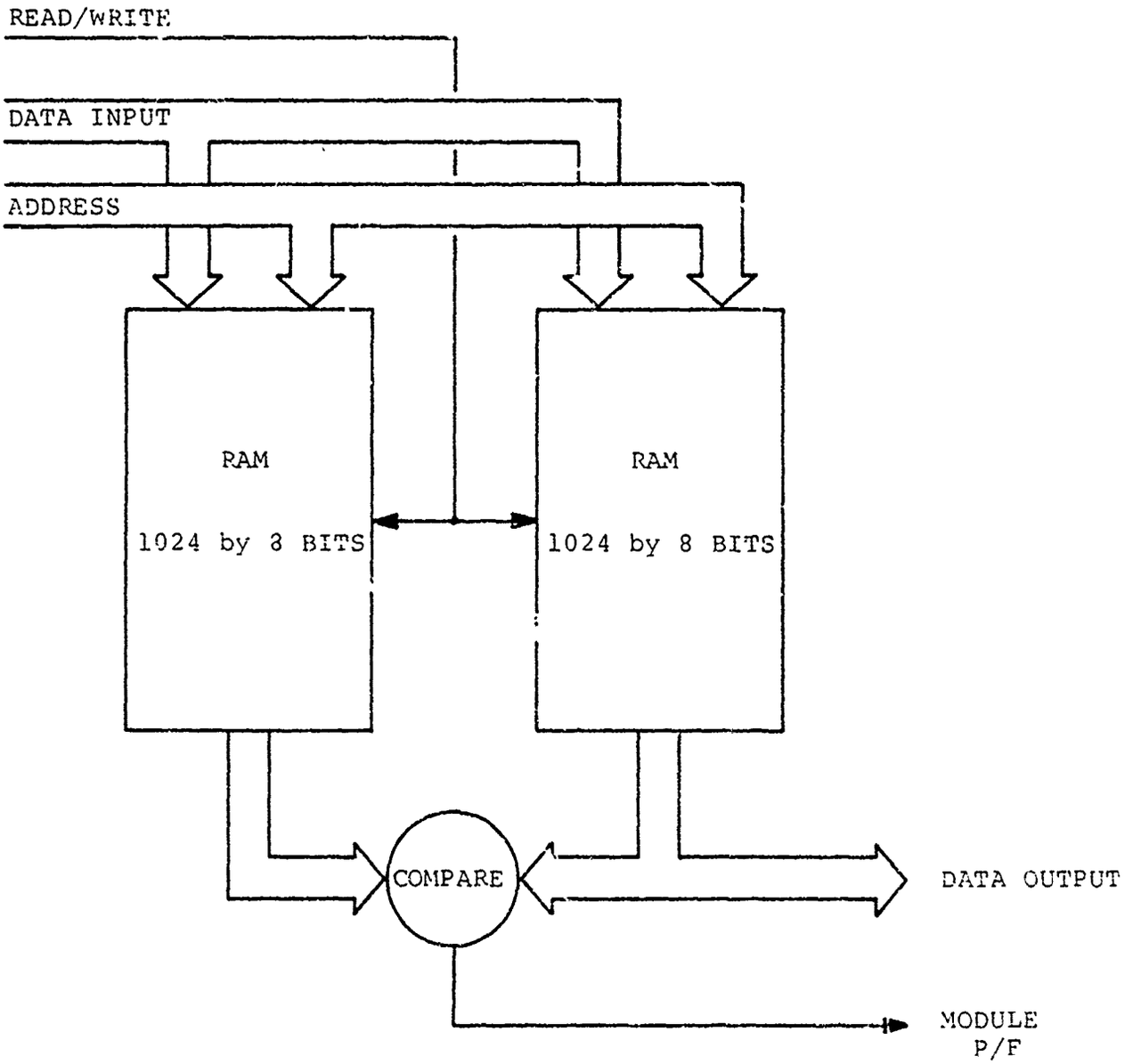



**FIGURE 3-7 RAM BIT BY DUPLICATION**

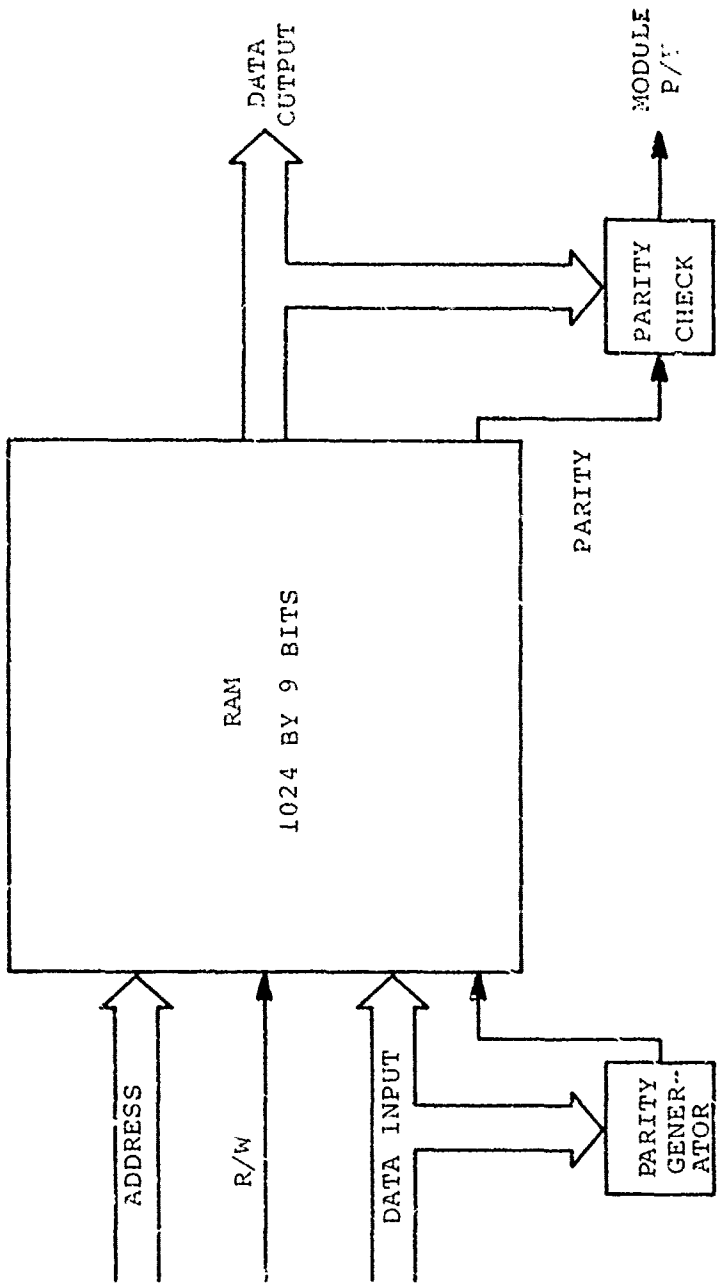



**FIGURE 5-8 RAM BIT BY PARITY**

of a double bit error, $P_\alpha$. Therefore,

$$P\alpha = 36 \cdot \left(\frac{1}{1024}\right) \cdot \left(\frac{1}{1024}\right) = 3.43 \times 10^{-5}$$

which means that of the possible errors in the memory locations only about three in 10,000 will be double bit errors.

One can readily see that the occurrence of more than two simultaneous, even number of failures is much less likely under the same assumptions. Thus multiple, even number of bit failures are not of major concern in parity checking approached for properly organized memories.

### 5.7.1.3 Single BIT Error Correction

It is possible to correct errors in memory systems with the use of multiple check bits. For single bit error correction the number of check bits required may be determined by the inequality:

$$2^k \geq m + k + 1$$

where m=number of data bits, and k=number of Hamming parity bits or check bits [6]. Solving this equation where m = 8, shows that a minimum of 4 check bits are needed to correct single bit errors in 8 bit data words. While it may appear that 4 bits to check 8 bits is a considerable amount of overhead, error correction can *improve* the MTBF. This is something that simple error detection can never do. Error correction works by generating and storing k parity bits from specific subsets of bits from the data word. When the data is read out, the parity generation process is repeated on the data bits. If these two sets of bits are different, an error has occurred. The bit pattern made by the exclusive OR-ing of the two sets of parity bits can be decoded to indicate in which bit position the error has occurred. This bit is complemented which corrects the error. Error correction as described cannot correct multiple bit errors and cannot detect most multiple bit errors. A block diagram illustrating this approach for a QED RAM module is shown in Figure 5 9.

### 5.7.1.4 Off-Line Testing

To provide built-in test in an off-line mode, it is necessary to provide a test pattern generator on the module. The most common type of bit pattern used to check a RAM is a "checker board" (alternating ones and zeroes) pattern which is easily generated. While a cyclic pseudorandom generator can also be utilized as an input sequence, a simple shift register can be used to generate an alternating one and zeroes test pattern with less hardware. To check for all stuck-at-faults, it is necessary to write and read back a one and a zero in each bit position so actually two tests patterns (one the inverse of the other) are needed. After the test pattern has been written into memory, the pattern generator is restarted and the bit pattern read from the memory is compared to the regenerated input bit pattern. Any pair of bits that are different indicates a fault. A block diagram of a RAM module with off-line BIT is shown in Figure 5-10.

In using test pattern techniques, it is necessary to provide a method of bus isolation within the module so that the test patterns are not propagated throughout the whole system. Other necessary functions are a clock, an address counter, and some basic control for the test mode operation. In all off-line testing that is performed while the system is operating, the system designer must make provisions for restoring the data in the tested modules so it will not be lost because of the test.

### 5.7.1.5 Recommendations

The recommended approach for the built-in test of the RAM modules is word parity. A block diagram of a module with word parity is shown in Figure 5-11. This method is recommended because it uses the least additional hardware while providing a high error detection capability. It is clear that the parity technique uses the least number of additional memory circuits, since the parity approach uses only one additional memory circuit, the error correction approach uses four, and the duplication approach uses eight. The off-line approach requires no additional memory circuits, but it does use a substantial number of MSI and SSI packages. Because of the strong correlation between the number of packages (especially complex memory circuits) and costs (board space, power and failure rate) it is of fundamental concern to minimize this parameter. The parity approach certainly satisfies this goal. The slight reduction in the fault detection capability of the parity approach is of little concern. Built-in-test using parity on the RAM modules can detect single bit errors in over 99 percent of the module's gates, which is near perfect by most any measure.

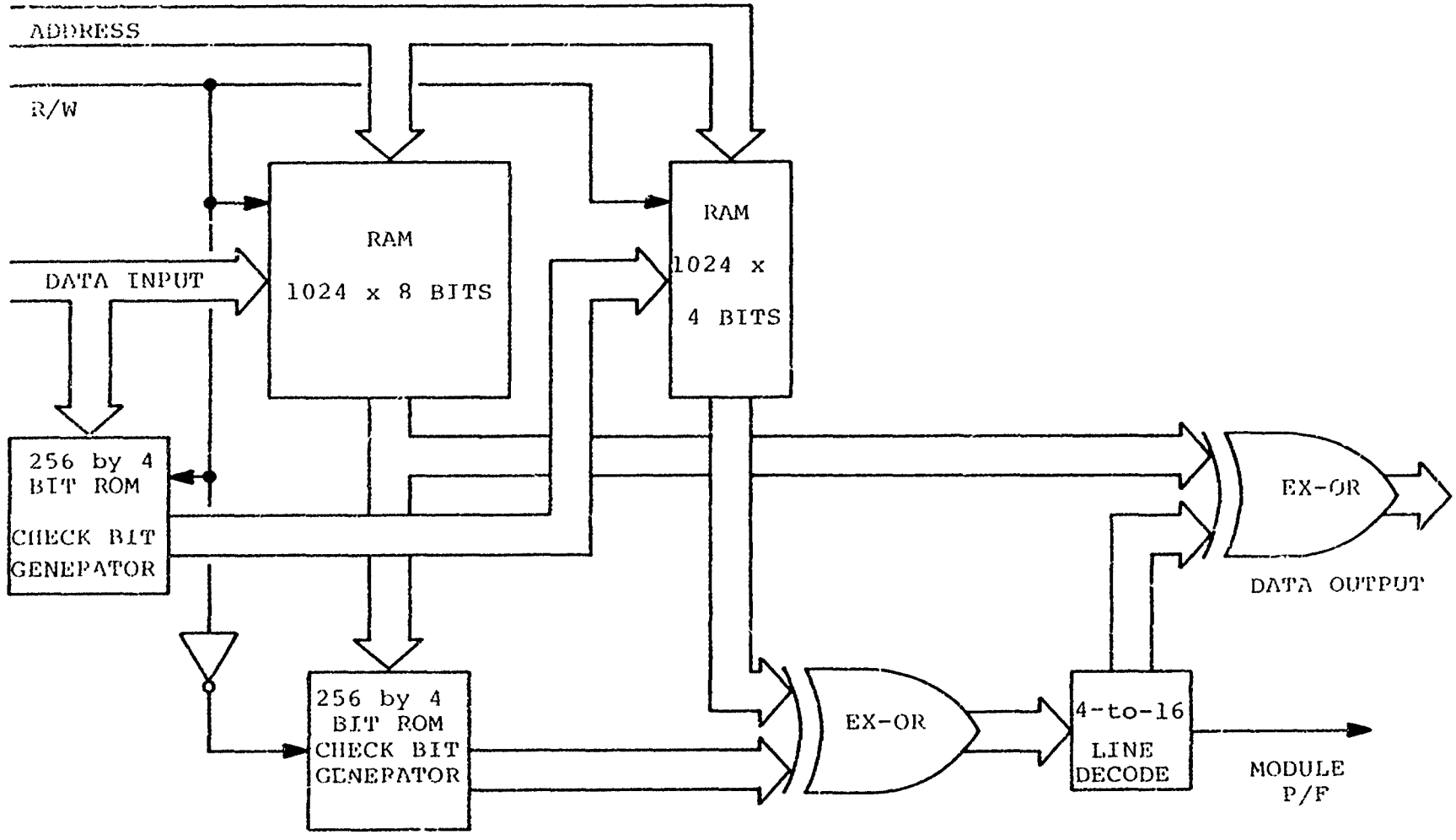



**FIGURE 5-9 RAM BIT WITH ERROR CORRECTION**

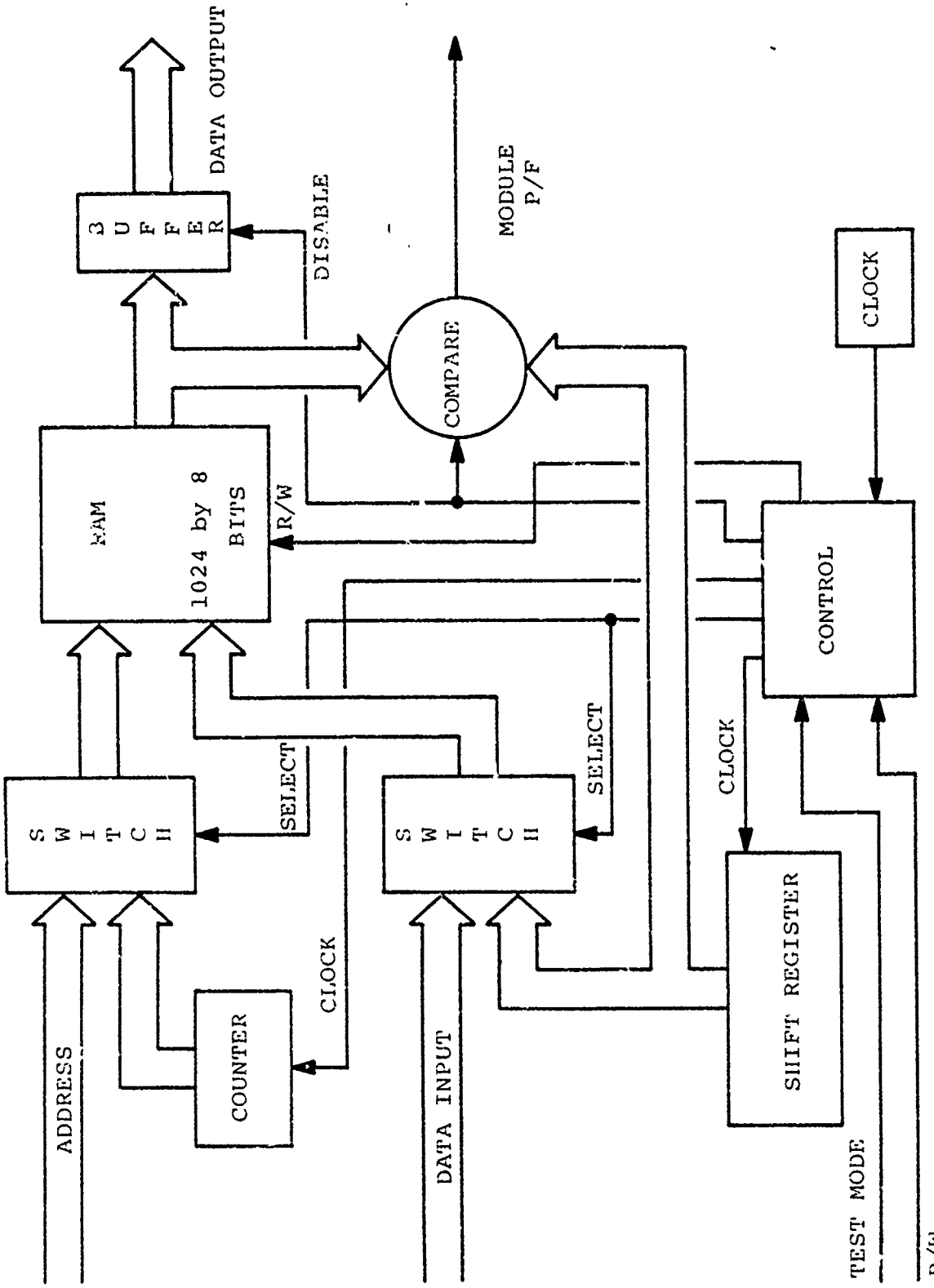



**FIGURE 5-10 RAM OFF-LINE BIT.**

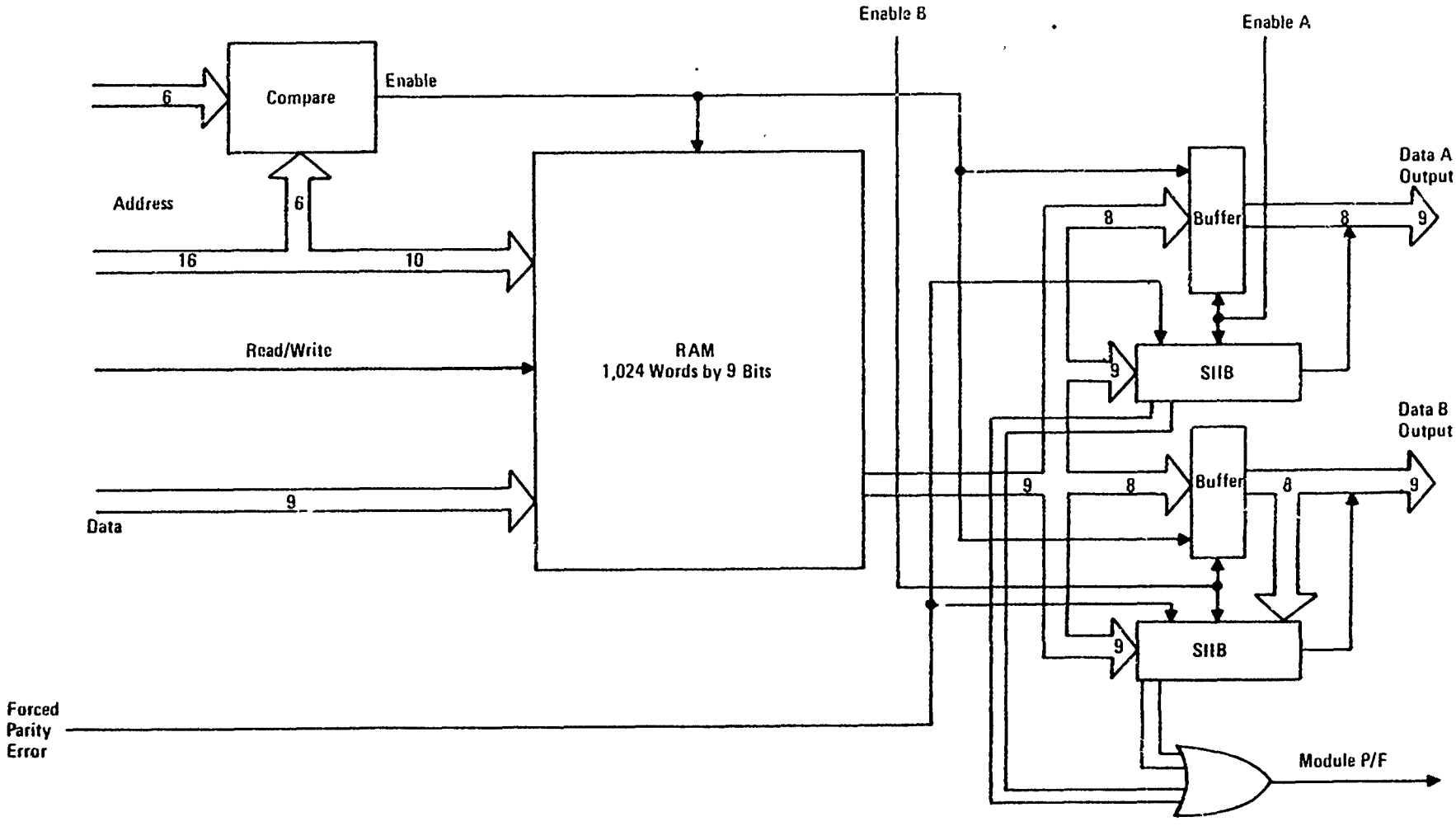



**FIGURE 5-11 RAM MODULE WITH WORD PARITY**

The particular drawbacks of the other BIT approaches will be further explained. The memory duplication technique is the most costly in terms of added circuitry and increased failure rate at the module level. Furthermore, at the system level, memory modules are often the most heavily used, so that simple module duplication would result in a very expensive system. Duplication offers no significant advantage over word parity. Therefore, the duplication technique is not recommended for any potential memory applications

In general, error correction techniques are desirable only if advantage is taken of the fact that the error has been corrected. Therefore, no repair/replacement is necessary when an internal error has been corrected. A module error then translates into a multiple bit memory error. In using the single bit error correction technique described for the RAM module, the module error detection capability is less than 25 percent for the double bit errors. This means that even though the module will operate properly for a greater length of time, there is a low probability that a module error will be detected. This then violates the basic BIT goals of high module error detection capability which is why this error correction method is not recommended.

The off-line mode of testing RAMs is a less practical BIT method for a number of reasons. Unless the checking is done prior to the initialization of system data, the contents of the memories under test must be temporarily moved to another location. While this protection of data is not difficult it is a potential source of error and further increases the time necessary to complete a test. Because of the potentially long period between testing times, errors are not quickly detected, reducing the confidence in the system outputs The biggest disadvantage of off-line testing is that it takes time away from the system. While not critical in all applications it is unlikely that system designers would welcome these restrictions and the added design effort needed to control the timing of the tests.

### 5.7.2 Read Only Memories

Read-only memories (ROMs) function in the same manner as RAMs during read cycles and can be tested using similar techniques. The on-line modes of providing BIT to RAMs are theoretically directly applicable to testing ROMs. A full description will not be repeated but there are a few modifications that must be noted. With respect to duplication of memory circuits, the only functional change from the RAM diagram is duplication of the package enable decoder so that each decoder can drive a separate ROM array. The coding and storage of the check bits for error correction is the same as the RAM except that the code is programmed in other ROMs and there is no code generation done on the module (only code checking).

A problem of geometry arises when implementing parity on a ROM memory in a method similar to a RAM memory. A RAM circuit may only be one bit wide; that is to say for each address only one bit may be read or written. To provide a memory with multiple bit words, a number of RAM circuits are used with their address lines bused together so that each RAM integrated circuit supplies one bit of the data word. To add a parity bit the designer simply adds one more RAM circuit to the address bus to store the parity bit for each data word. This point is illustrated in Figure 5-12.

In contrast to the RAM case, typical commercially available ROM circuits are either four or eight bits wide, so that for each address four or eight bits are output at one time. Since ROMs are not commercially available in one bit wide (or nine bit wide) configurations, the addition of a ROM in a manner like the RAM is impossible. Parity can still be added in an efficient manner as shown in Figure 5-13. The specific arrangement of the ROM circuits on the QED module is four circuits, each having 512 eight-bit wide words. Four enable lines are decoded from the address so that only one of the four circuits provides the output of the module. Parity can be added to the module with the addition of a 512 by 4 bit ROM. The nine address lines that are connected to the four data ROMs will also be connected to the parity ROM. The single parity bit for each data word can easily be decoded from the 4 available from the parity ROM by a select 1 of 4 decoder

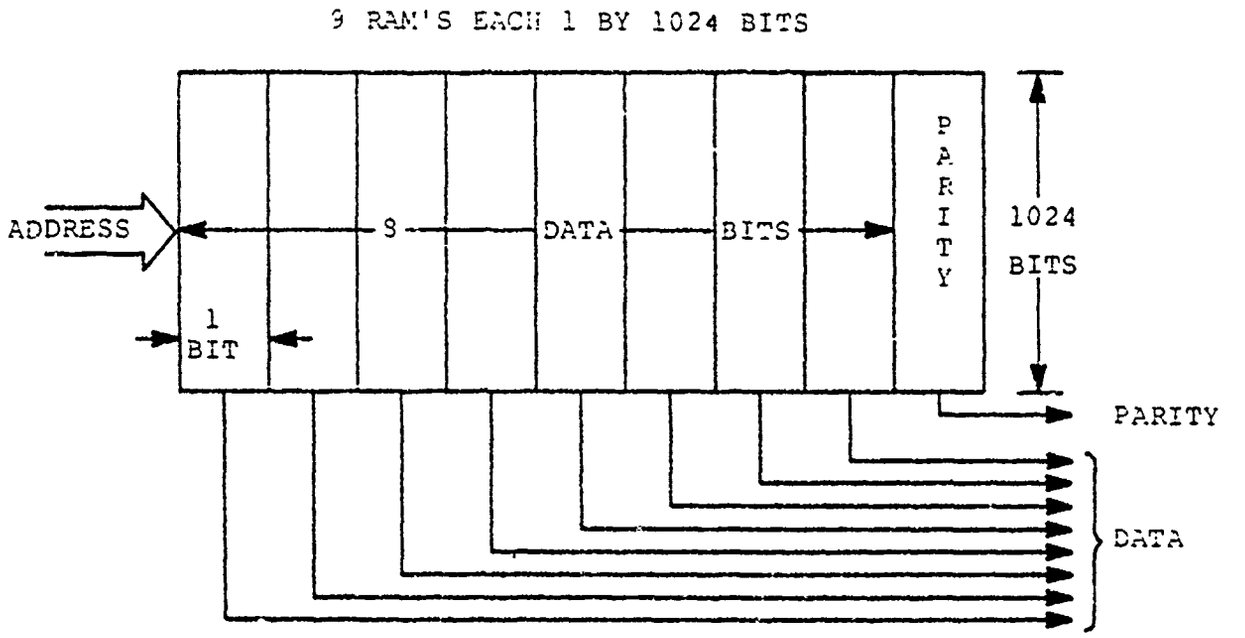



**FIGURE 5-12 PARITY IMPLEMENTATION IN RAM**

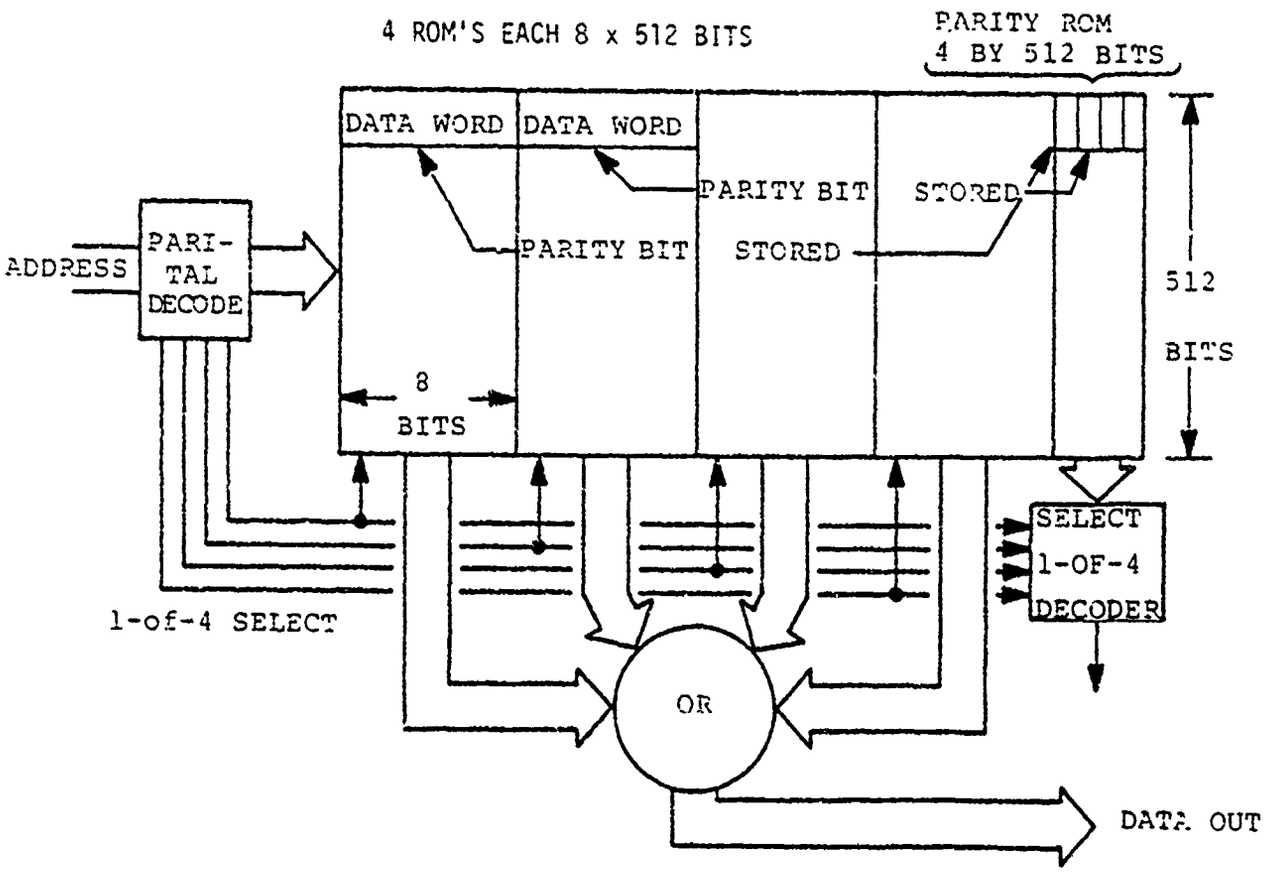



**FIGURE 5-13 PARITY IMPLEMENTATION IN ROM**

connected to the circuit enable lines. Using this technique all ROM storage is utilized and the additional BIT circuitry is not excessive.

The off-line technique described for use in the RAM module is not suitable for use with the ROM module because the testing circuitry cannot write data into the ROM. However, another off-line approach, longitudinal parity, can be used to provide BIT for the ROM modules. This is achieved by stepping through each address of the ROM and calculating the parity on each bit position, using eight modulo-2 counters, a longitudinal parity word is generated which can be compared with a fixed constant to verify the correctness of the data in the ROM. The fixed constant would most economically be stored in one of the ROM locations, so that the 2048 word ROM module would become a ROM module with 2047 usable locations. A diagram of this approach is shown in Figure 5-14.

The recommended BIT approach for the ROM module is word parity and a block diagram of such is shown in Figure 5-15. The methodology for selecting the best BIT method is identical to the RAM module. The word parity implementation provides a check on almost the entire memory module with a relatively small increase in hardware. This BIT technique can detect all single bit stuck-at-faults within the ROM integrated circuits and some internal addressing decoder errors. While this approach cannot detect all addressing errors, it will generally detect half of them. This reduction in error detection capability affects the overall fault detection capability only in an inappreciably way.

The major drawbacks of each of the other BIT methods will now be considered. The major disadvantage of the off-line approach is the large number of additional packages necessary for BIT. This does not affect the failure rate as much as may be expected because the added packages are relatively simple and therefore have low failure rates. (However, the parity approach has a lower failure rate than the off-line approach.) Another drawback to this BIT approach is that the testing function and system use cannot occur at the same time. However, it is possible to perform the testing, a part at a time, during the period that the system is not using the module (i.e., cycle stealing). With proper system timing, in many applications it is possible to check the ROM module and not slow down system throughput. However, this control of timing would make the system more complicated. The error correction technique is not desirable because of the low module error detection capability as explained for the RAM module Duplication offers no significant benefits to BIT, it only increases cost.

### 5.7.3 First-In-First-Out Memory

The FIFO memory is similar to the other memory modules when considering built-in test. The RAM on-line error detecting techniques have an analogous method in the checking of the FIFO. Memory duplication with a comparator checking the outputs can be applied to the FIFO just like the RAM. The coding techniques also can be applied to the FIFO. In fact, provisions for parity are made by the manufacturers of the FIFO in that the circuits are designed to store none-bit data words. This fact makes parity checking the most economical of all BIT approaches because a very small amount of additional hardware is needed. A block diagram of the FIFO module with parity is shown in Figure 5-16.

An off-line approach that functions much like the RAM BIT is applicable to the FIFO. Using this approach, a test control signal enables a test pattern generator which writes into the memory until it is full. The memory then outputs the information which is compared to the output of the test pattern generator. Using this scheme, the system designer must use caution in the timing of the test mode signal because in order not to lose any data, the test mode signal should be given only when the FIFO is empty.

The FIFO module is best suited to use parity as a built-in-test method because the integrated circuits are made to handle nine bit words. This fact makes other built-in test methods much more costly (with only slight

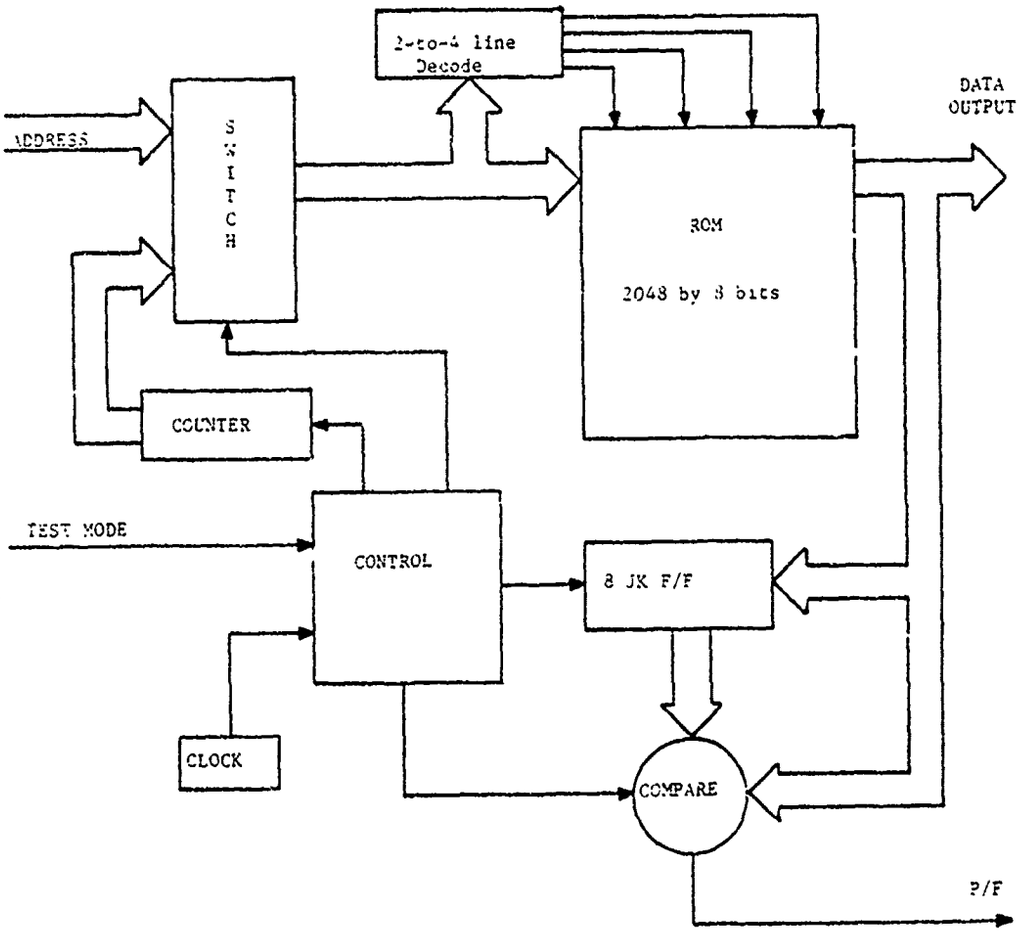



**FIGURE 5-14 ROM BIT BY LONGITUDINAL PARITY**

gain in error detection capability) because of the higher number of additional circuits necessary to implement them. Additional circuits also increase the failure rate and power consumption of the module.

### 5.7.4 Further Discussion of Built-In Error Correction Techniques for Memory Class Module

It is envisioned that functional digital modules like the QED family will be used in applications where manual intervention to effect system repair may not be possible. For example, tactical fighter aircraft and helicopters require highly reliable electronic systems for relatively short duration missions. In these applications it is of little consequence to be able to note that an electronic module has failed, if action to repair that fault cannot be taken. Fortunately, in the case of digital memory functions, relatively simple codes exist which can be used to automatically correct errors in a straightforward way without operation intervention. This discussion considers a particular error correction code and its application to memory class functions.

To take full advantage of error-correction codes and achieve minimum maintenance and built-in-test, it is necessary to provide detection capability for double bit errors in addition to the correction of single bit per word errors. Since the single bit errors are only errors within the modules, the operation of the module as far as the system is concerned, is faultless. Therefore the module need not be replaced when a single error has

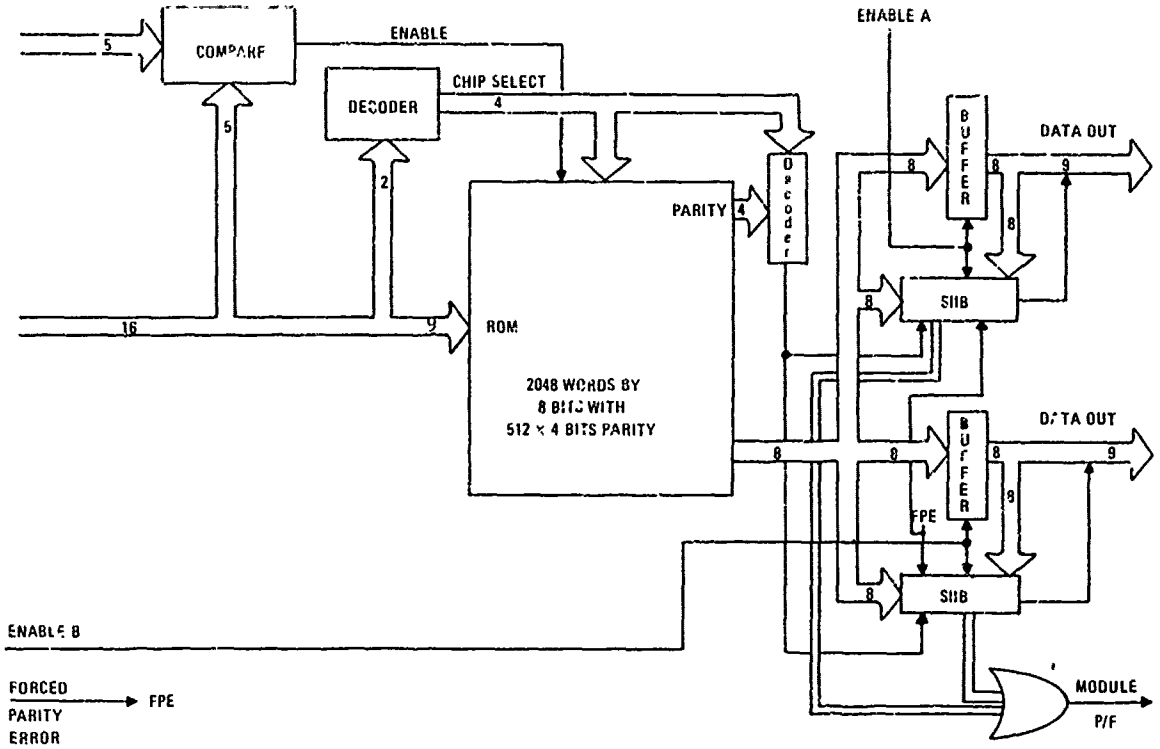



FIGURE 5-15 ROM MODULE WITH WORD PARITY

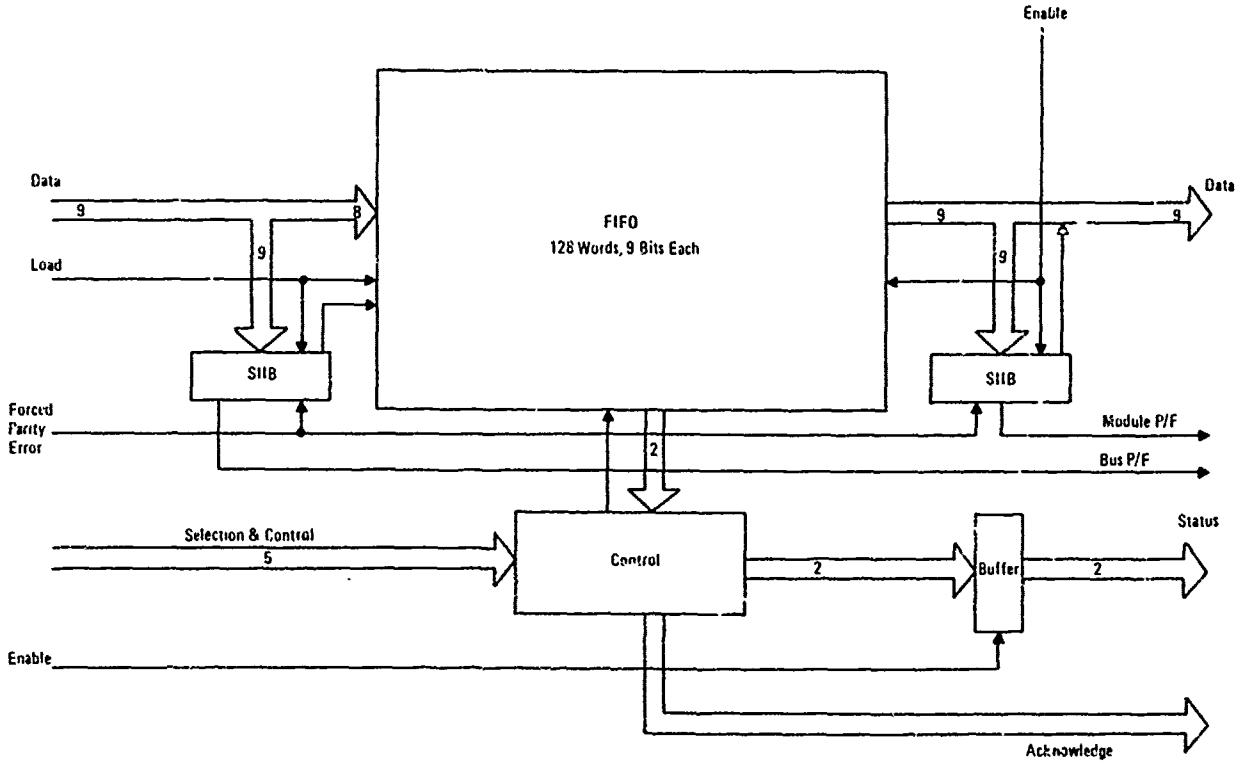



**FIGURE 5-16 FIFO MODULE WITH PARITY**

occurred. It must be replaced only when a double error has occurred. In this way the MTBF of the module can be increased, but only if double-bit errors can be detected. In general it is possible to convert a single-bit error correcting code into a single-bit error correcting with double bit error detecting code by adding one additional bit which is a parity bit over all of the data and code bits. Using the Hamming single-error correcting code described previously, five check bits would be needed to provide single error correction with double error detection on an eight bit data word. The coding of the check bits would be identical to the single error correction case (with the addition of the overall parity bit) as would be deciding and error correction. Double errors are indicated by the overall parity of the word being correct, but the error correcting check bits indicating an error has occurred. Single (correctable) errors are indicated by an error in both the overall parity and the error correcting codes indicating an error. No error, of course, is indicated by no errors in any of the parity checks.

To more easily quantify the increase in MBTF in order to justify the additional cost in space, power, and dollars needed to implement an error correcting code, it is necessary to make several assumptions. The single error correcting, double error detecting code will correct all single bit per word errors and detect all double bit per word errors in the storage elements. The following analysis will assume all errors are independent. A single bit per word parity scheme as previously described will be used as a basis for comparison. The parity approach will detect all single bit word errors and will assume that these errors are independent. The differences in detection/correction of three or more errors will be ignored because of the small probability of their simultaneous occurrence.

The following data for each type is based on a module storing 1024 eight bit words using 1K by 1 bit RAMs for storage and MSI/SSI level support circuitry. In the computation of the reliability of the memory using error correcting codes, the memory module is divided into two parts. One part contains the memory chips themselves and their reliability is computed based on the fact that only 12 of the 13 memory chips need to be working in order for the whole module to be working. For this portion of the module the reliability does not follow the exponential model (failure rate independent of time) that most electronics do, but rather is described by the equation

$$R_c = 13e^{-12\lambda t} - 12e^{-13\lambda t}$$

where $R_c$ is the reliability of the array of memory chips, $\lambda$ is the failure rate of one memory chip and t is time.

The other portion of the memory module is made up of all the other circuitry in the module. This portion of the module must be completely operational for the memory module to function properly. The reliability model for this portion of the module is the common exponential decay based on a uniform failure rate, (the failure rate is independent of time) and can be expressed as

$$R_S = e^{-\lambda_S t}$$

where $R_s$ is the reliability of the support circuitry, $\lambda_S$ is the composite failure rate of the circuitry (i.e., the sum of the failure rates of all the individual support chips) and t is time.

To compute the reliability of the entire memory module, the reliability of the memory chips is multiplied by the reliability of the support circuitry. This gives a reliability equation of the form

$$R_m = R_c R_s = 13e^{-(12\lambda c + \lambda s)t} - 12e^{-(13\lambda c - \lambda s)t} \qquad \text{(Equation 1)}$$

where $R_m$ is the reliability of the entire memory module, $\lambda_c$ is the failure rate of one memory chip, $\lambda_s$ is the failure rate of all of the support circuitry and t is time. The reliability of the memory modules with parity is

determined in the conventional manner and can be expressed as

$$R_p = e^{-\lambda_p t}$$

where $R_p$ is the reliability of the module with parity, $\lambda_p$ is the composite failure rate of the module (the sum of the failure rates of the individual ICs) and t is time.

Figure 5-17 shows the reliability as a function of the time for the QED RAM module for the two built-in test approaches just described (parity and single error correction with double error detection). The important question in evaluating the error correcting memory module is to quantify the gain in failure rate. With this information, it is possible to make a well founded decision on whether the additional costs of error correcting are worth the increased reliability and reduced maintenance.

Because the reliability curve of the error correcting memory module is not of form $R = e^{-\lambda t}$, it is impossible to assign a single number to the failure rate. However, it is possible to define a factor, $F_R$ (that is a function of time), that realistically indicates the improved reliability achieved using the error correcting technique. This factor, $F_R$, can be defined as:

$$F_R(t) = \frac{\ln(R_p)}{\ln(R_{ec})}$$

where $R_p$ is the reliability (at a given time) of the module with parity and $R_{ec}$ is the reliability (at the same given time) of the module with error correction. For the case of single error correction on an eight-bit data word this factor is expressed as

$$F_R(t) = \frac{-\lambda_p t}{\ln[13e^{-(12\lambda_c + \lambda_s)t} - 12e^{-(13\lambda_c + \lambda_s)t}]} \qquad \text{(Equation 2)}$$

where the variables are the same as defined in equations 4.8 and 4.9.

The derivation of this equation can be briefly described. To define the failure rate gain the desired factor, $F_R$, should be the quotient of the failure rate of the module with parity, $\lambda_p$, and an analogous failure rate, $\lambda_a$, that satisfies the equation

$$R_m = e^{-\lambda_a t} \qquad \text{(Equation 3)}$$

and solving for $\lambda_a$

$$\lambda_a = \frac{-\ln(R_m)}{t}$$

where R and t are corresponding values that satisfy the error correction reliability equation 1. In other words, a failure rate that would give the identical reliability (at a given time) as the error correction curve. This factor must be a function of the amount of time the module is in service since the error correction reliability curve and the parity reliability curve are not of the same form. As previously described in this paragraph, $F_R$ was defined as

$$F_R(t) = \frac{\lambda_p}{\lambda_a(t)}$$

Substituting equation 3 into this equation yields:

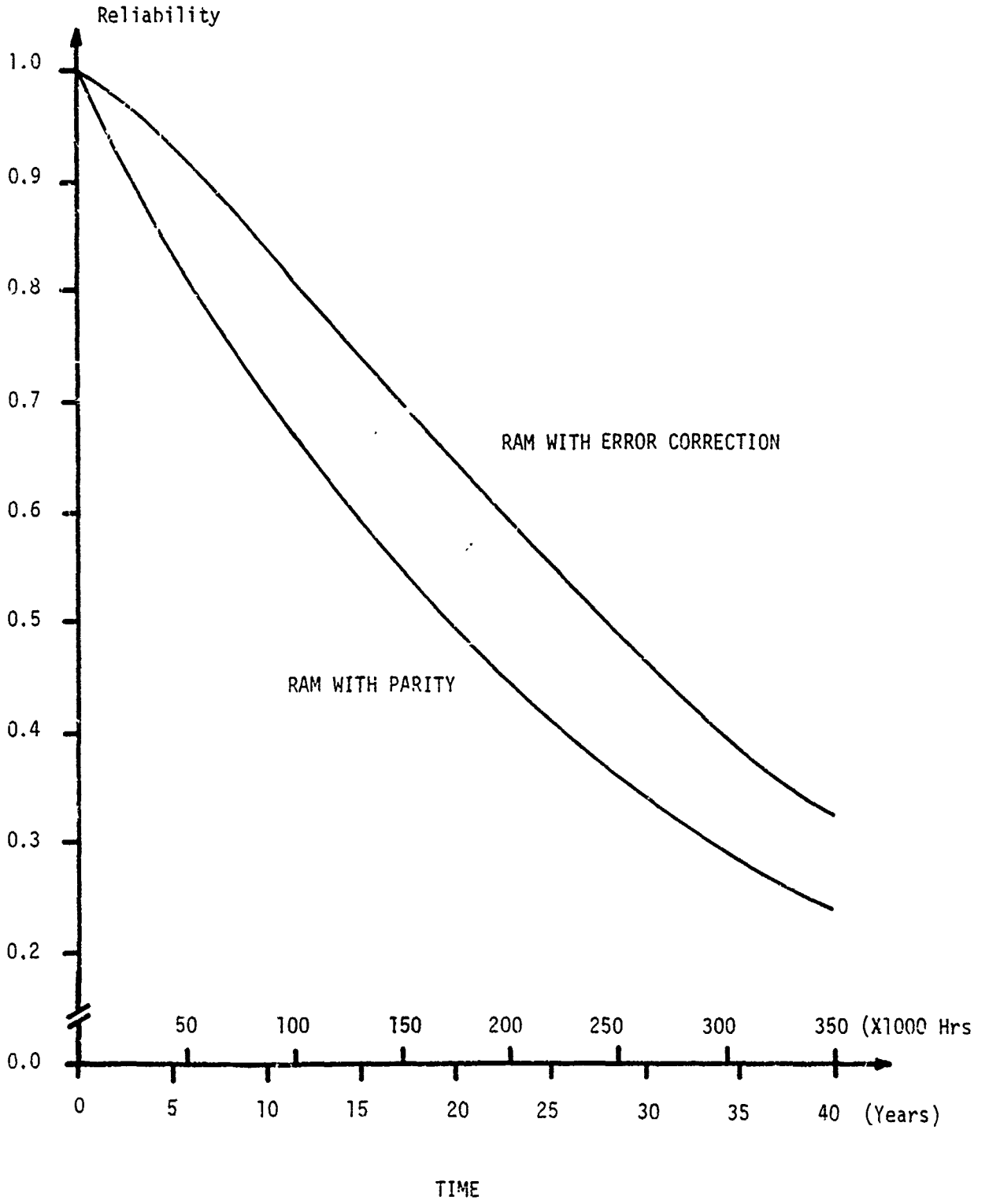



**FIGURE 5-17 RELIABILITY OF RAM MODULES**

$$F_R(t) = \frac{-\lambda pt}{\ln (R_m)}$$

which is identical to equation 2 once equation 1 for $R_m$, the reliability of module with error correction, is substituted into it.

Figure 5-18 shows this increased reliability factor $F_R$ as a function of time for the QED RAM modules as previously described. One can see that from an initial failure rate gain of a little over three, the reliability increase falls to a gain of about two after ten years. This shows the substantial reliability increase that can be achieved for quite a number of years. In addition, through preventative maintenance, it is possible to keep this failure rate gain over three. This achievement is made simply by an annual replacement of the modules that have internal failures that are being corrected. This replacement reduces the probability of a double bit error (and, therefore, increases the reliability of the module) because of the modules that have already suffered single bit error are removed.

### 5.7.5 Standard Interconnection and Interface BIT

A special circuit has been designed to provide the parity generation and checking necessary to implement the recommended BIT technique. In addition, it was designed to provide parity generation and checking for the interconnecting data buses of all the QED modules. This capability provides a BIT technique to check the module input and output circuits as well as to detect wiring and connector faults. This special circuit is called the Standard Interface and Interconnection BIT (SIIB). The SIIB provides a module level, on-line (concurrent) fault monitoring capability which can supply module pass/fail information to a system fault monitor.

The SIIB, as a standard BIT circuit, was designed to be used on a large number of different modules. Because of its multi-function design, it can be used in alternate ways to check different circuits, thus reducing the number of necessary standard BIT circuits. In particular, the SIIB is designed to provide an error detection capability for the input latches and output buffers of the QED modules. In addition, the SIIB provides an error detection and isolation capability for interconnecting circuitry including logic card connectors and backplane wiring.

On certain QED modules the SIIB also provides part of the error detection capability for the functional portion of the module as in the case of the Random Access Memory (RAM) module. The BIT functional approach embodied in the SIIB, is to check parity on incoming data and to check and generate parity for outgoing data. The following section presents a detailed functional description of the SIIB.

#### 5.7.5.1 Functional Description

A block diagram of the SIIB circuit is shown in Figure 5-19. It consists basically of two 8-bit odd parity generators and checkers. This BIT circuit is used to check incoming data parity and to verify the performance of the input latches as shown in Figure 5-20. When used in this manner the $P/F_a$ line indicates the results of a comparison of the parity of A Data and the transmitted parity bit. A failure of the comparison indicates a fault in either the logic card connector or in the intermodule connecting wiring. The $P/F_b$ line indicates the results of the comparison of the parity of the output data of the latches and the parity of B Data. Thus, $P/F_b$ verifies the proper operation of the input data latches. For timing purposes, these comparisons are made only when the latch is enabled. This approach not only indicates when errors occur in the interconnection wiring and the input latches, but it facilitates fault localization by distinguishing between such faults.

In general, the parity-out line is not used. However, in the memory modules and certain I/O modules the parity-out data can be used to check the functional portion of the module. In this case, the SIIB circuit takes the place of a parity generator. Also, in this application, the SIIB aids fault localization.

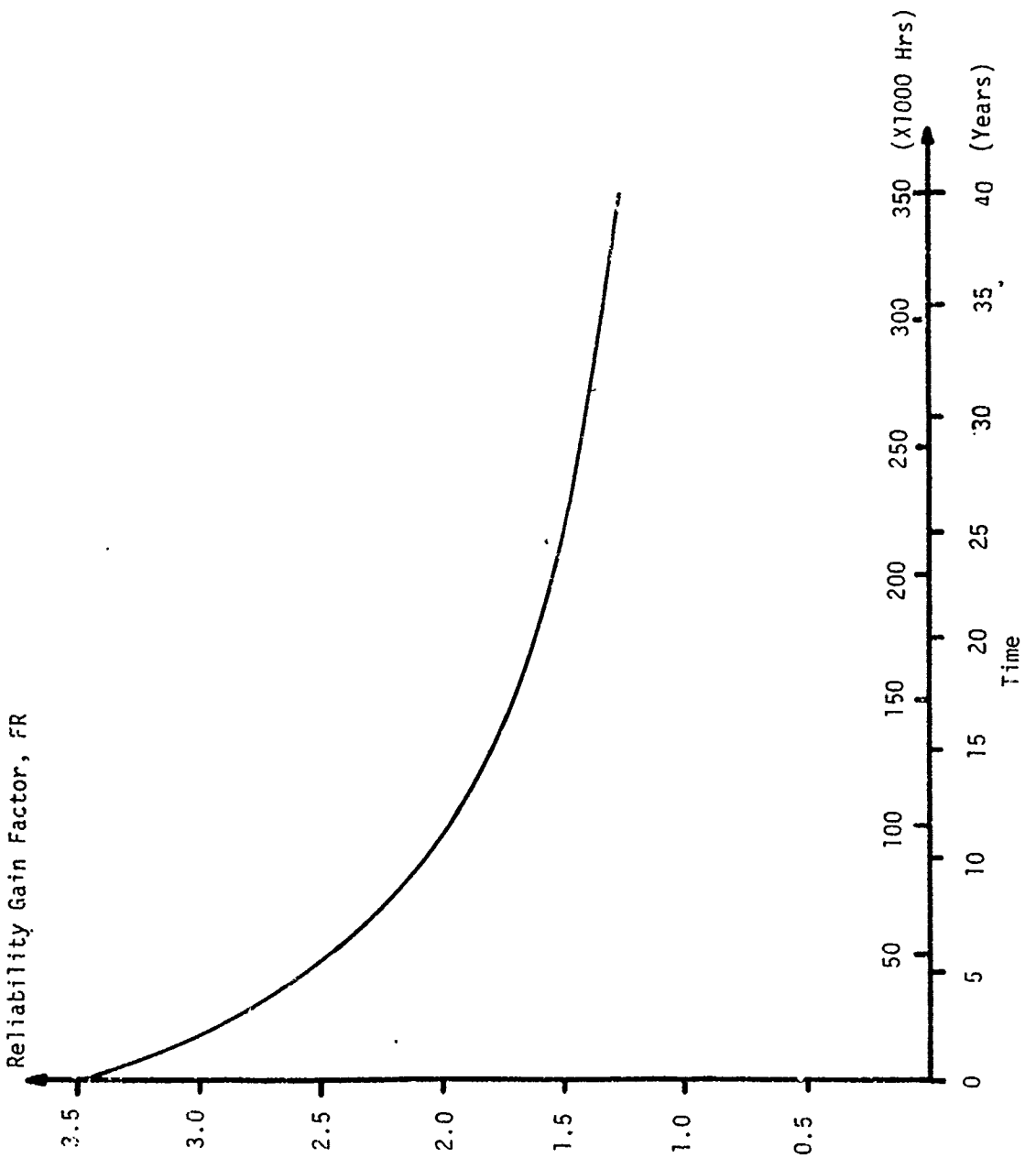



**FIGURE 5-18 RELIABILITY GAIN USING ERROR CORRECTION**

The SIIB circuit can also be used to check the output buffers as shown in Figure 5-21 When applied in this manner, the $P/F_A$ line in general has no meaning and is, therefore, left unconnected. The $P/F_B$ line gives an indication of the comparison of the data parity bits on each side of the output buffer. This comparison is made only when data is enabled out. A failure of the comparison denotes a failure in the output buffers. In addition to checking the output buffers, the SIIB generates the parity bit to be transmitted on the bus along with the data bits.

In order to verify the proper operation of the SIIB, a Force Parity Error (FPE) capability is included as a part of the SIIB circuit. The FPE line can be used to check not only the SIIB itself but the associated wiring and subsystem error detection/localization monitor hardware and software. When the FPE line is driven high, and the subsystem error detection/localization equipment does not indicate an error, there is a fault in the BIT circuitry By driving the FPE line high, both P/F lines should go high indicating an error. If the P/F lines do not respond, the SIIB has a fault within it and should be replaced. If the P/F lines are correct and the subsystem does not indicate a fault, the subsystem or the interconnecting wiring is in error.

### 5.7.5.2 Logic Diagrams

Figure 5-22 is a gate level logic diagram of the SIIB. The B-parity output is a tri-state output to allow it to drive a bus when used for output checking. The pass/fail lines are normal TTL outputs and the FPE and Enable lines are normal TTL inputs. The delay to the output enable is present to prevent the P/F lines from indicating a failure for the very short time before the circuit reaches a "steady state" value. This delay may be accomplished by cascading standard TTL inverters. The SIIB circuit as shown contains 22 equivalent logic gates. Should a custom MSI chip be fabricated, it can easily be made to fit in a standard 24 pin dual in-line package.

### 5.7.5.3 Truth Tables

Table 5-1 is a logic truth table for the SIIB showing the relationship between the circuit inputs and outputs. To make the table more readable and without sacrificing generality, the data inputs are represented by only the parity bit word in this table.

### 5.7.5.4 Circuit Implementation

The SIIB may be implemented with currently available off-the-shelf small and medium scale integrated circuits. An alternate approach would be to design a single custom MSI chip to realize the SIIB function. The characteristics of each of these implementation alternatives are given in Table 5-2. Although either option is theoretically possible, in a practical sense the single custom circuit is by far the most viable. Not only is the package count drastically reduced, but so is the failure rate (0.065 versus 0.187 failures in $10^6$ hours) and the power dissipation (50 mw versus 400 mw, assuming it will be made in a low power Schottky version).

It should be noted that the proposed standard BIT circuit is a candidate high volume IC. That is, due to its universality, it may be used in a wide variety of applications and therefore can be produced in large quantities. The resulting advantage, of course, is that high volume ICs tend to be very inexpensive and candidates for multiple sourcing.

### 5.7.5.5 Critical Parameters

The only potentially critical SIIB timing problem involves the length of the delay between the input enable and the pass/fail output data valid period. The delay must be long enough to allow the latches and buffer outputs to become valid and for the propagation delay within the SIIB to take place. By delaying the outputs as indicated in Figure 4.16, the P/F lines never give "false alarm" spikes while the data is becoming stable.

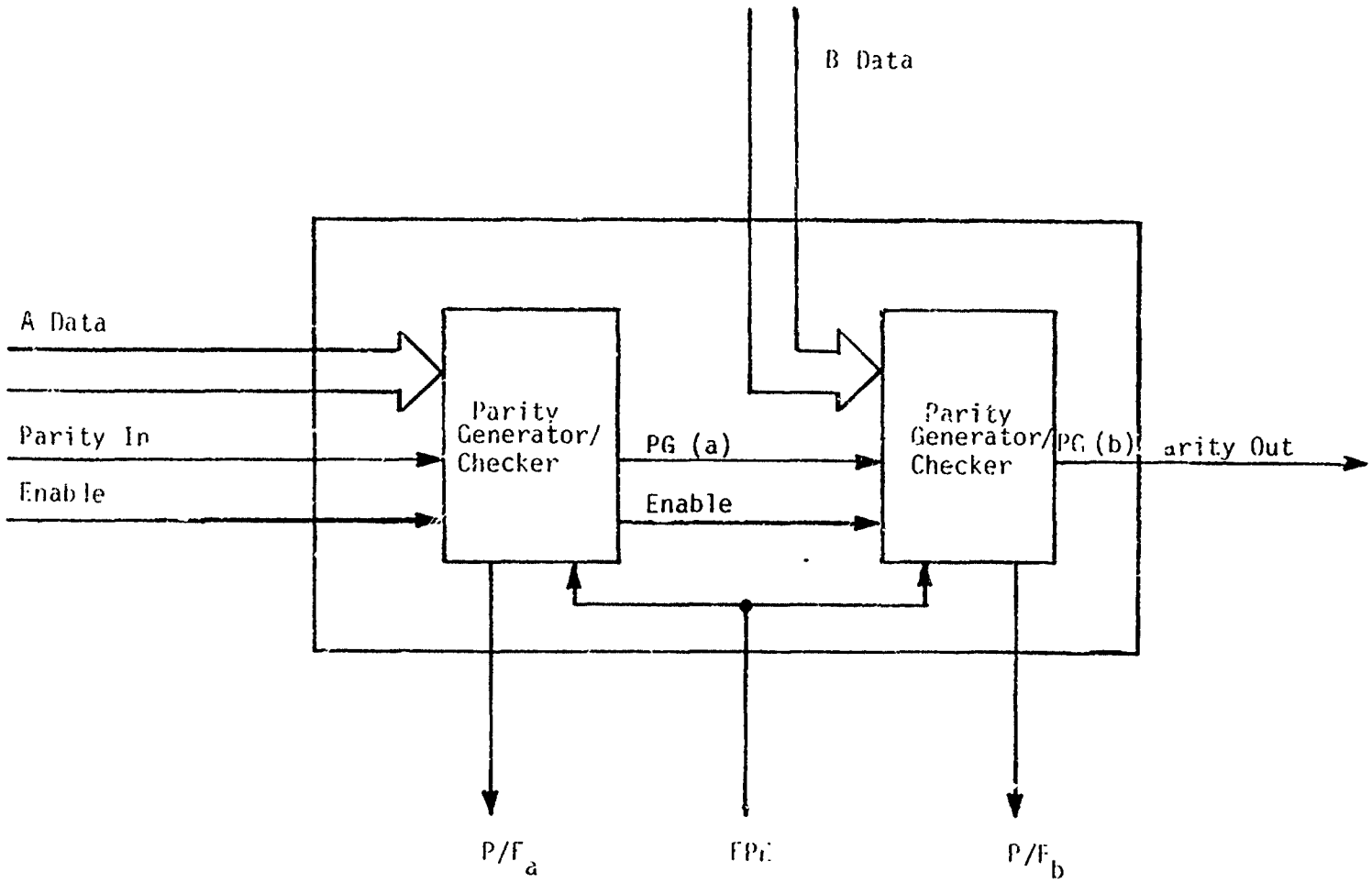



**FIGURE 5-19 SIIB FUNCTIONAL BLOCK DIAGRAM**

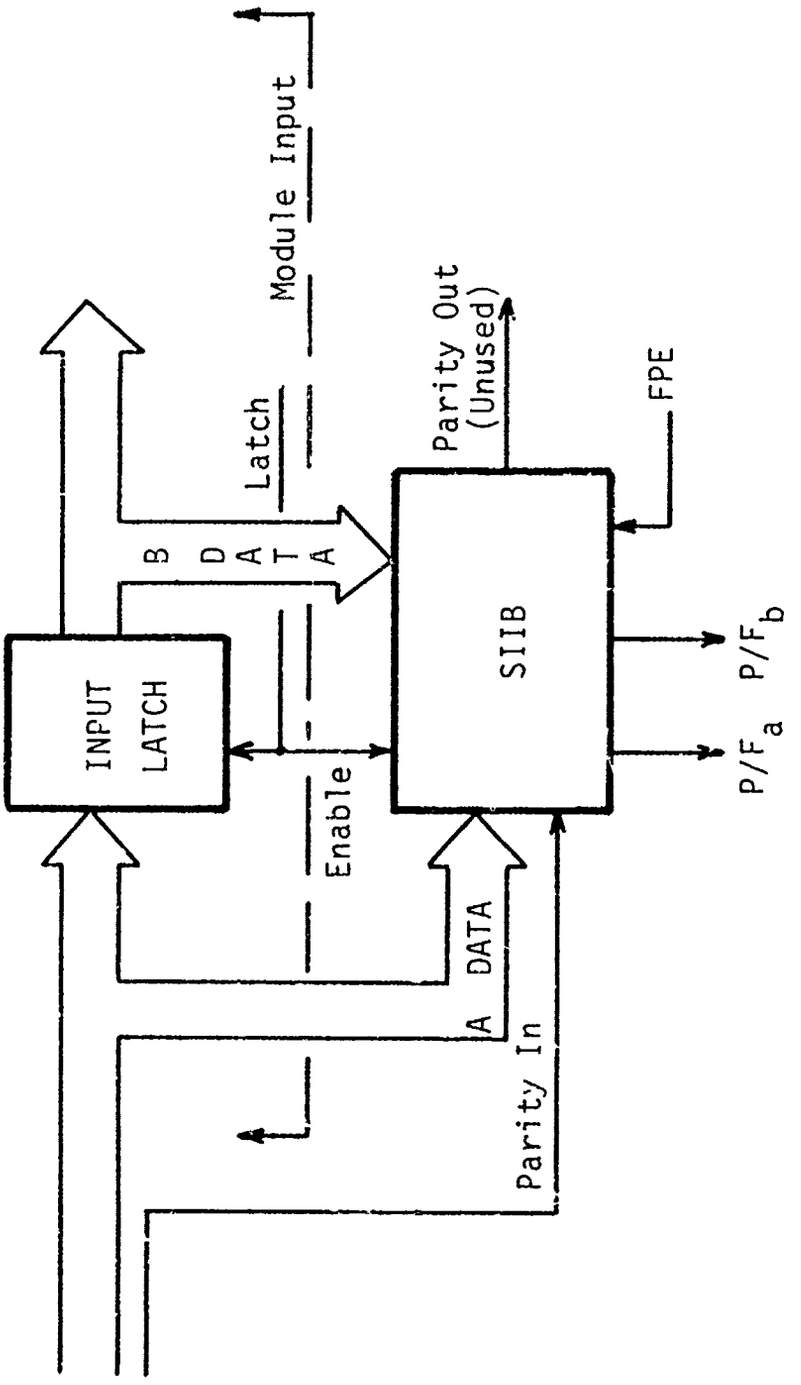



**FIGURE 5-20 INPUT CHECKING**

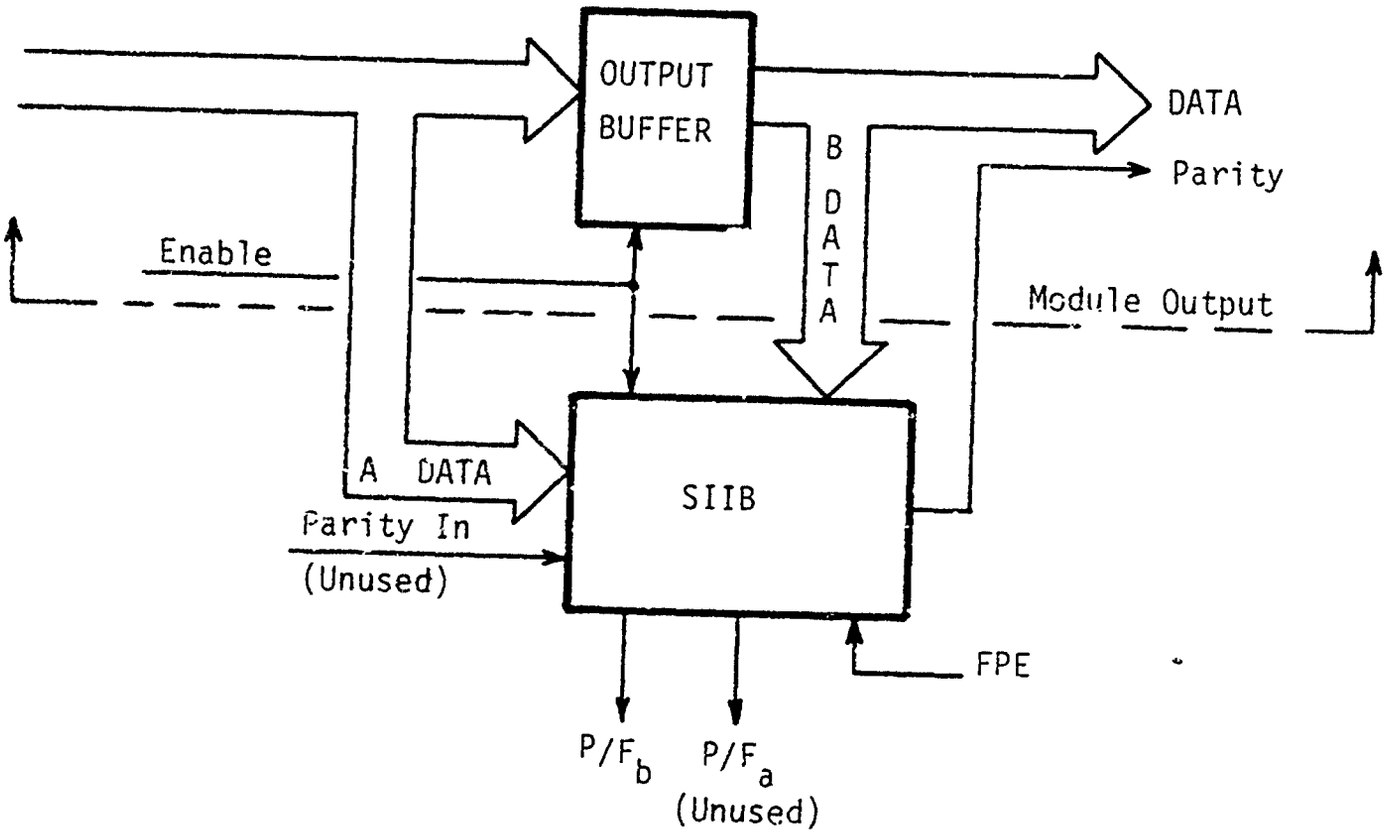



**FIGURE 5-21 OUTPUT CHECKING AND PARITY GENERATION**

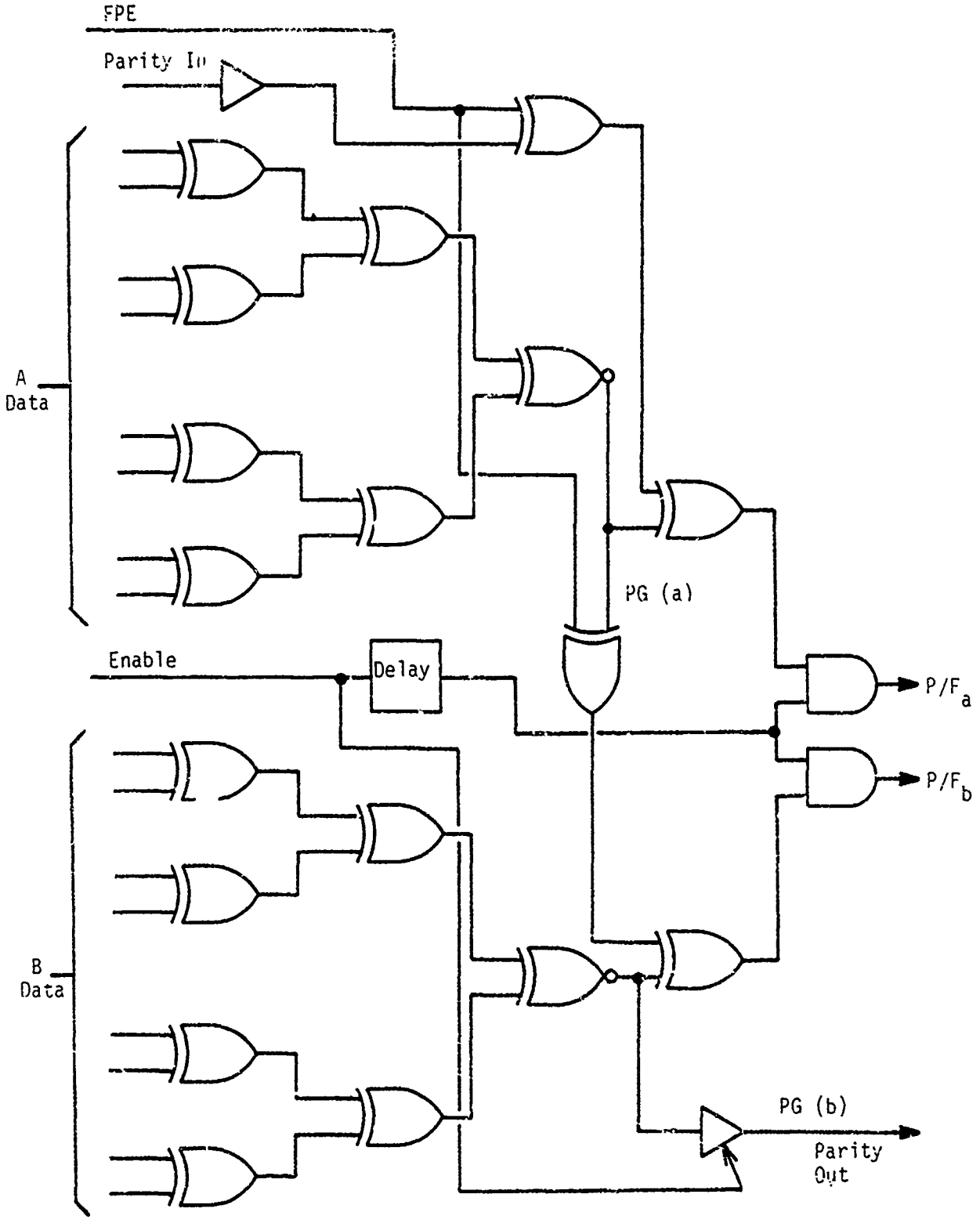



**FIGURE 5-22 SIIB LOGIC DIAGRAM**

| Enable | FPE | Number of Input Data Bits High (0 = Even, 1 = Odd) Data Word A | Data Word B | Parity Bit In | $P/F_b$ | $P/F_b$ | Parity Out |
|---|---|---|---|---|---|---|---|
| 0 | X | X | X | X | 0 | 0 | H |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

1 = High

0 = Low

X = Don't care

H = High Impedance State

**TABLE 5-1 TRUTH TABLE FOR SIIB**

| SIIB | Implementation Using Existing MSI | Implementation Using Custom MSI |
|---|---|---|
| No. of Gates | 48 | 22 |
| Packages | 4 | 1 |
| No. of Pins | 14 pins/package | 24 |
| Failure Rate | $0.387/10^6$ Hours | $0.265/10^6$ Hours |
| Power Dissipation | | |
| Typical | 750 mw | 50 mw |
| Max | 1200 mw | 90 mw |

**TABLE 5-2 SIIB IMPLEMENTATION ALTERNATIVES**

The SIIB circuit adds no critical timing requirements to any of the operations of the modules. This results from the fact that the SIIB is basically a monitor and causes no processing to start or stop. It is recommended that the SIIB be implemented within an integrated circuit technology such as low power Schottky which minimizes gate loading and power consumption.

### 5.7.5.6 QED Module Test Equipment Requirements

The recommended SIIB circuit is self-checked through the use of the Forced Parity Error (FPE) input. When this input is driven high and the module is enabled, the P/F outputs indicate a fault. It is recommended that the module input and output SIIBs be checked separately by enabling only the input latch and then enabling only the output buffer. It is necessary to have a valid data word on the input of the SIIB when performing these tests. This simply means that data supplied to the module under test must be valid.

The QED module test equipment must also perform a check of the parity output. Actually, two tests are necessary to determine proper operation for a data word with both even and odd parity. Proper operation is defined by Table 4-1. On output buffers, the high impedence state must be verified by not enabling the output and performing standard electrical checks on the parity output. This testing can be performed simultaneously with the electrical testing of the output data. This would involve an additional process but no new procedure would be required.

**5.8 Fail-Safe Design:** It is desirable that all fault and BIT circuitry be of a fail-safe design. Two examples of implementing fail-safe circuitry are shown in Figures 5-23 and 5-24. Figure 5-23 is a fail-safe circuit which is used with liquid-coolant flow switches, air-flow switches, pressure sensors, and temperature sensors. It operates so that if either the flow is interrupted or the connector J1 or P1 is disconnected (or a wire breaks) the appropriate BIT fault-indicator lights.

In Figure 5-24, a fail-safe presence detector is shown which is fail-safe when coupled with BIT automatic test for either a loss of the crystal detector or a break in the coaxial circuit. In normal operation, the circuit requires the presence of a diode signal and a "0" level going pulse at the output of AR1 to prevent the tripping of the flip-flop. This, coupled with a BIT automatic test (during which there is no output from the diode, the AR1 output is a logic "1", and there is a BIT look-gate signal and a BIT amplified input RF signal) results in tripping the flip-flop circuit and creates a low at the output of the driver. This low is then sent to the BIT NAND gate and ANDed with other fault-circuit signals to obtain a BIT automatic test signal.

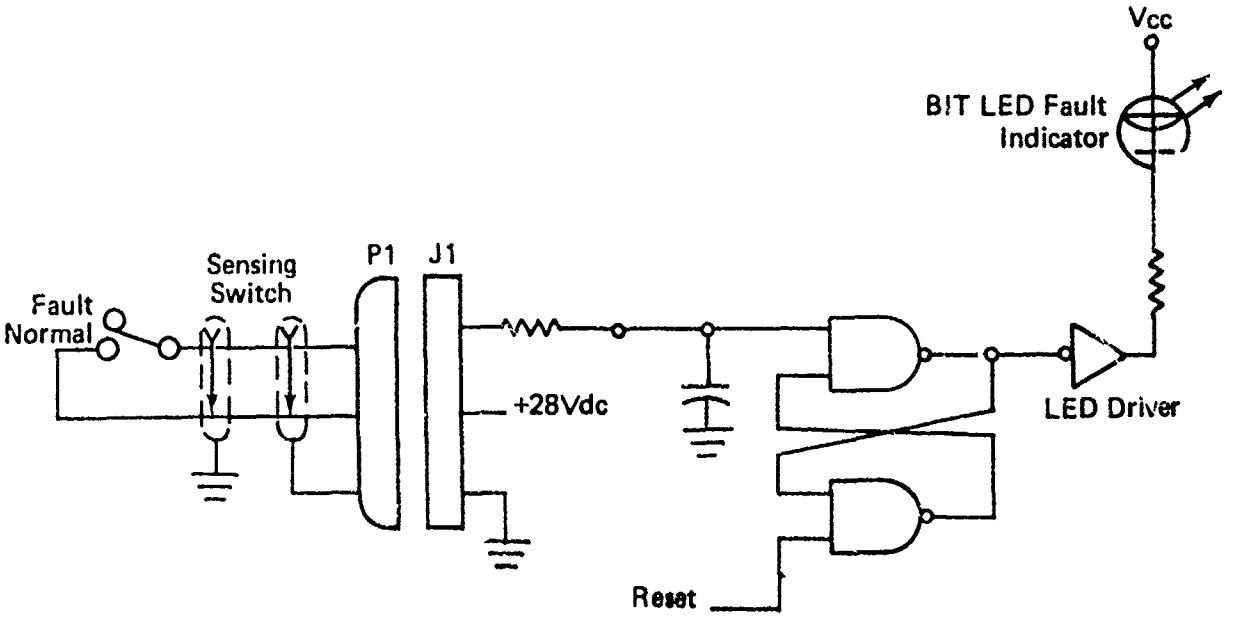
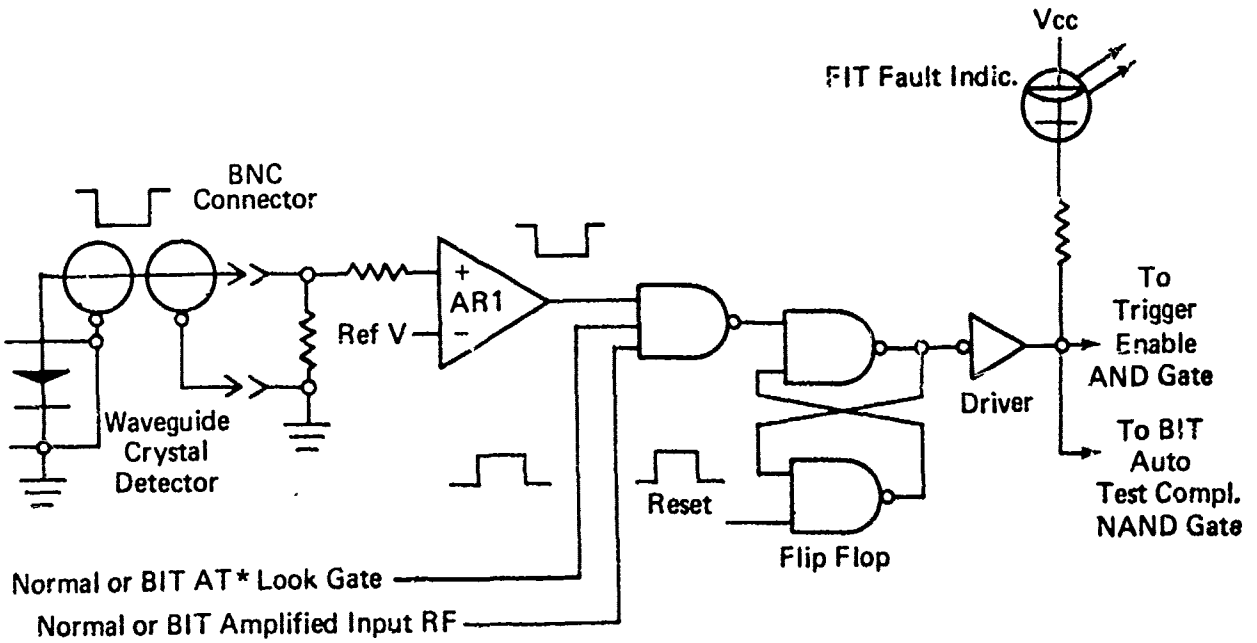



**FIGURE 5-23 TYPICAL FAIL-SAFE SENSOR-SWITCH CIRCUIT**




* AT =Automatic Test

**FIGURE 5-24 FAIL-SAFE PRESENCE DETECTOR**

# SECTION 6

# DATA USED IN BIT DESIGN

In order to arrive at an optimum design for BIT, the designer requires a great deal of information concerning design requirements, reliability of discrete circuits, and effects of interfacing with other systems. This section defines the data necessary to arrive at an optimum design approach for BIT.

## 6.1 Worst-Case Stress Analysis

The worst-case analysis should investigate the stresses on each system component over the specific temperature ranges and end-of-life tolerances. It is required to assure equipment (including BIT) operation under all combinations of component tolerance and supply-voltage variations and to ensure that no circuit failure occurs due to emergency conditions or power failure. Computer programs for worst-case analysis are available.

## 6.2 Nominal Stress Analysis and Reliability Prediction

Nominal stress analysis is necessary to determine the component failure rate, calculate the system MTBF, and ensure suitability of components for the design under consideration. Part-failure rates can be derived from MIL-HDBK 217B.

BIT analysis depends upon the failure rates of individual components within the system. The reliability prediction for those cards detected by BIT can now be separately identified from those cards not detected by BIT. The BIT detectability-level determination can then be computed, as will be discussed later.

In the calculation of $M_{ct}$, other parameters to be determined are the time to disassemble, interchange, reassemble, and checkout the system. This is a major item in the mean-time-to-repair analysis. Once this is known, the time remaining for isolation and localization is then available. With this data, the designer can make a decision as to what level of BIT is required to locate the malfunction to a functional area and how small that functional area should be.

## 6.3 Failure Mode And Effect Analysis

Another analysis of great importance is failure mode and effects analysis (FMEA). This assists the designer in assuring that malfunctions will not adversely affect other systems and it affords a check to determine the exact level of BIT in the system. Degraded modes of operation are described so that acceptable degraded modes are not classified as failures.

FMEAs are derived from functional flow diagrams, schematics, and timing-sequence diagrams and are then used to identify components affected by the failure modes. For each failure mode, a list of components and their associated failure rates are compiled. The high failure probability components are then chosen as BIT interrogation candidates. FMEAs are usually performed down to the replaceable module level of the system. Fail-safe precautions are recommended when a critical function is found to depend upon a single equipment. Alternate modes of operation and practical means of overriding automatic functions in the event of failure are often discovered via this process. The objective of these studies is the definition of design problems and critical areas which must receive corrective action. Figure 6-1 depicts the elements of an FMEA.

In order to perform an FMEA, the first requirement is to establish the basic performance, safety,

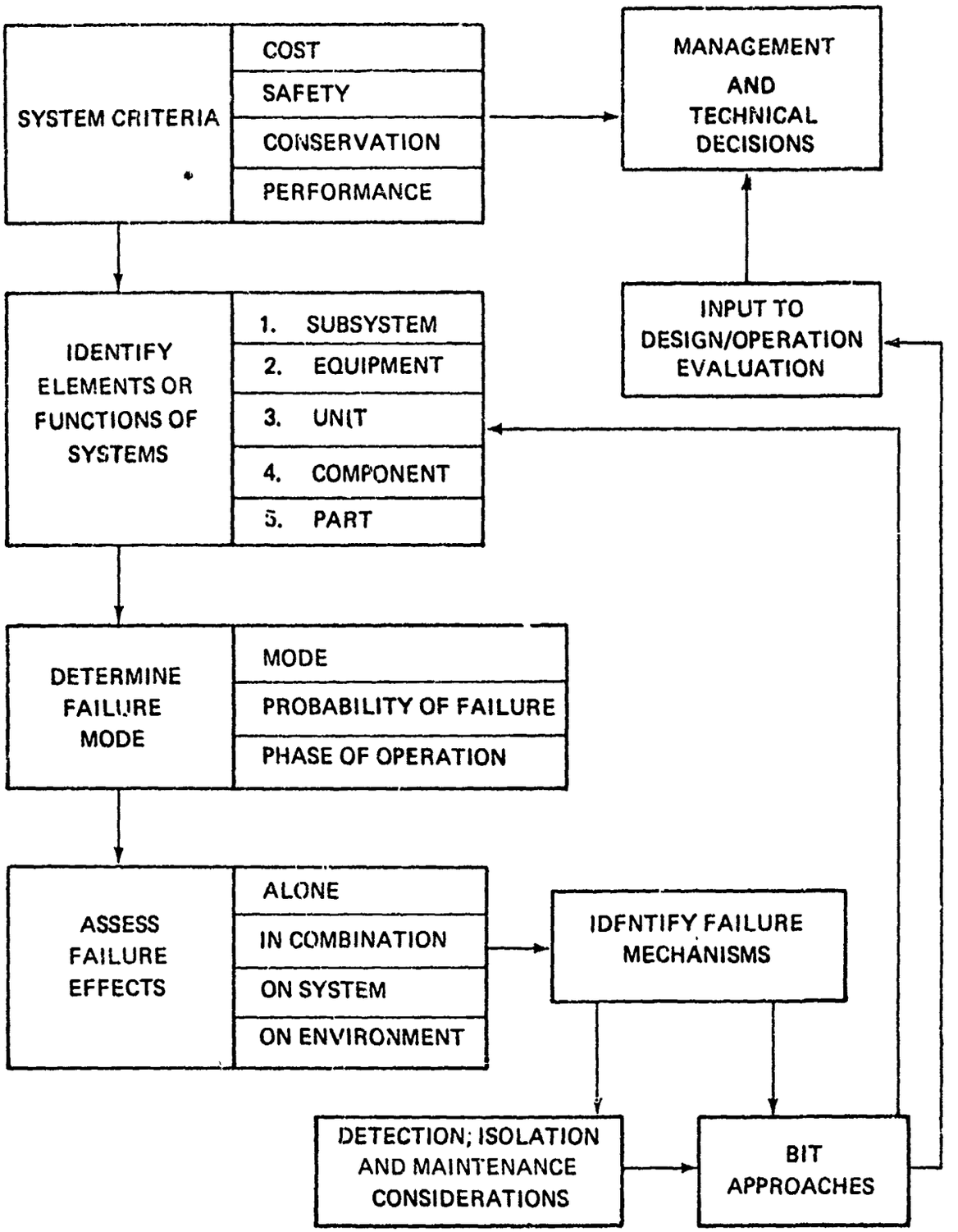



**FIGURE 6-1 ELEMENTS OF A FAILURE MODES AND EFFECTS ANALYSIS (FMEA).**

maintenance, and inspection criteria for overall evaluation and to identify the elements or functions of a system, subsystem, or component to the appropriate levels which have a bearing on these criteria either directly or indirectly. This is accomplished by establishing detailed equipment and functional-block diagrams of the weapons system and its operation. The first-level breakdown for a radar is, for example, at the subsystem level (exciter, transmitter, antenna, etc.). Each subsystem element can be subdivided into the equipment which makes up the subsystem.

The equipment may be further subdivided into the units which make up the equipment; the units are then in turn subdivided into components. Finally, the individual parts of the system are defined. For each of the identified elements or functions, the failure modes will be determined. For example, an antenna-drive linkage may fail by separation or by binding.

For each failure mode thus defined, the effects can then be determined. The effects must be considered in three categories: the effect of the failure by itself without consideration of other related components or functions; the effect of the failure in combination with other elements of the system or other functions so as to determine if there is a compounding or mitigating results, and the effect of the failure on the total system operation. It is this identification of the failure effects in an ordered and logical manner which provides the ability to evaluate the systems operation in terms of safety, preventive maintenance, corrective maintenance, and periodic inspection.

Weapons system failure modes can be analyzed to determine the causes and mechanisms which result in a critical system failure. For example, in the case of a computer, it may be by a drop in line voltage resulting in a missing word. The missed word constitutes the effect. The identification of failure mechanisms, together with the criticality of the failure, are used to evaluate the system weaknesses of a specific design. A typical FMEA for an acoustic sensor is shown in Table 6-1.

Experience indicates that the most common failure mode of a digital component is an open circuit. Metallization breakdown and bonding defects within the device and external corrosion of contacts and solder joints are the contributing factors. Shorts are less frequent failure causes. These are due primarily to contamination within the device and are weeded out during the initial burn-in or infant-mortality phase.

Failures due to changes in component parameters are, fortunately, very infrequent. These can produce very distracting failure symptoms. For example, a change in output-drive capability due to a reduced transistor current gain can lower the noise margin of a signal line so that particular transmission patterns are improperly detected or crosstalk from neighboring lines may interfere. These pattern-sensitive problems are insidious and can be minimized by careful attention in the initial design to wire routing, power distribution, and component loading.

## 6-4 BIT Functional Block Diagram And Flow Analysis

Certain design data must be made available in order to fully evaluate a BIT philosophy. The most critical of these is a block diagram which shows not only the basic signal flow, but also the number of test points in each functional block. An example of BIT circuitry functional block diagram is shown in Figure 6-2.

All critical paths should be clearly noted on this block diagram. The functional blocks, defined by the block diagram, should be the same ones used in the worst-case and failure modes and effects analysis. This is to say that there should be a one-to-one correlation between functional blocks as defined on the block diagram and sections of the failure modes and effects analysis and the worst-case circuit analysis. This will allow the designer to correlate the information from the analyses, to be able to ascertain the effects of malfunctions upon system performance, and to determine which areas of the system require the greatest level of BIT.

System: Avionic

Subsystem: Acoustic Sensors

Equipment: Interface Unit

FAILURE MODE AND EFFECTS ANALYSIS

| Item Identification | | Stage of Oper. | Component System Function | Failure Mode and Cause | $\lambda \times 10^6$ | Failure Effect On | | | Failure Detection Method | Corrective Action And Safety Factor Available | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Nomenclature and Part No. | Item Code | | | | | Component or Functional Assy | Aircraft System | Total Aircraft | | | |
| Flutter Compensation 527-2001 | | TF | Provide Flutter Compensation Reference Signals To FM Demodulation (Playback) Signal Conditioners | Open Circuit Connection or Component Failure | 1.0 | No Flutter Compensation For FM Signal Channels | Sonobuoy Stored Retrieved Data Slightly Degraded At Low Frequencies | None | Appraisal Of Playback By Operator. | None | Previously Stored Data Not Impared |
| Electronics Power Supply (532-2001) | | TF | Regulates Power From Transport Power Supply To Provide +12 VDC For Signal Conditioning Circuitry In All Data Channels | Open Circuit Connection Or Component Failure | 29.5 | All Signal Conditioning Capabilities Lost. Output Voltage Exceeds Allowable Limits | ATR Function Lost For Remainder Of Mission. Data Storage And Retrieval Capabilities Degraded | None | GPDC Interrogation. Head Current. +5. +12, -12 Voltages | Shutdown Of ATR Indicated Via GPDC to Terminate Non-Operating Power On Condition | Previously Stored Data Not Impared |
| Electronics BITE (518-1001) | | TF | Monitors Electronics Unit Operating Parameters For Reporting GO/NO-GO Status | Open Circuit In Circuitry Involving One Or More Parameters. Short Circuit In Circuitry Involving One Or More Parameters | 15.0 | Loss Of Capability For Monitoring The Parameters Effected. False Failure Indications For Parameters Affected | Loss Of Status Reporting On Parameters Affected | None | None | None | Performance Of ATR Unimpared These Two SRA's Comprise A Single Functional Network. |
| MAD/GPDC Read/ Write (519-100D) | | TF | [illegible]tions Input [illegible]als For [illegible]ing And [illegible]g Back GPDC Interface Signals | Data Signal Conditioning Characteristics Fall Outside Design Limits–Connection or Component Failure | 2.8 | Conditioner Record Signal Falls Outside Specified Limits | MAD or GPDC Record Function Lost For Remainder Of Mission (Depending On Which Two Modules Are Affected) | None | GPDC Interrogation (If Failure Puts Read Write Current Outside Specified Limits) | None | Previously Stored Data Not Impared. |

**TABLE 6-1 ACOUSTIC SENSOR FMEA EXAMPLE**

The designer should have at his disposal a list of all interface requirements for all inputs and outputs. This list should include nominal and tolerance values for voltages, currents, and impedances. The designer should also know the availability of system multiplex inputs and outputs, memory locations, and spare or redundant circuitry.

The information just discussed constitutes the minimum data required for a system BIT design. The more data which can be obtained and which relate to system performance, system requirements, and hardware availability, the more effective the BIT design can be.

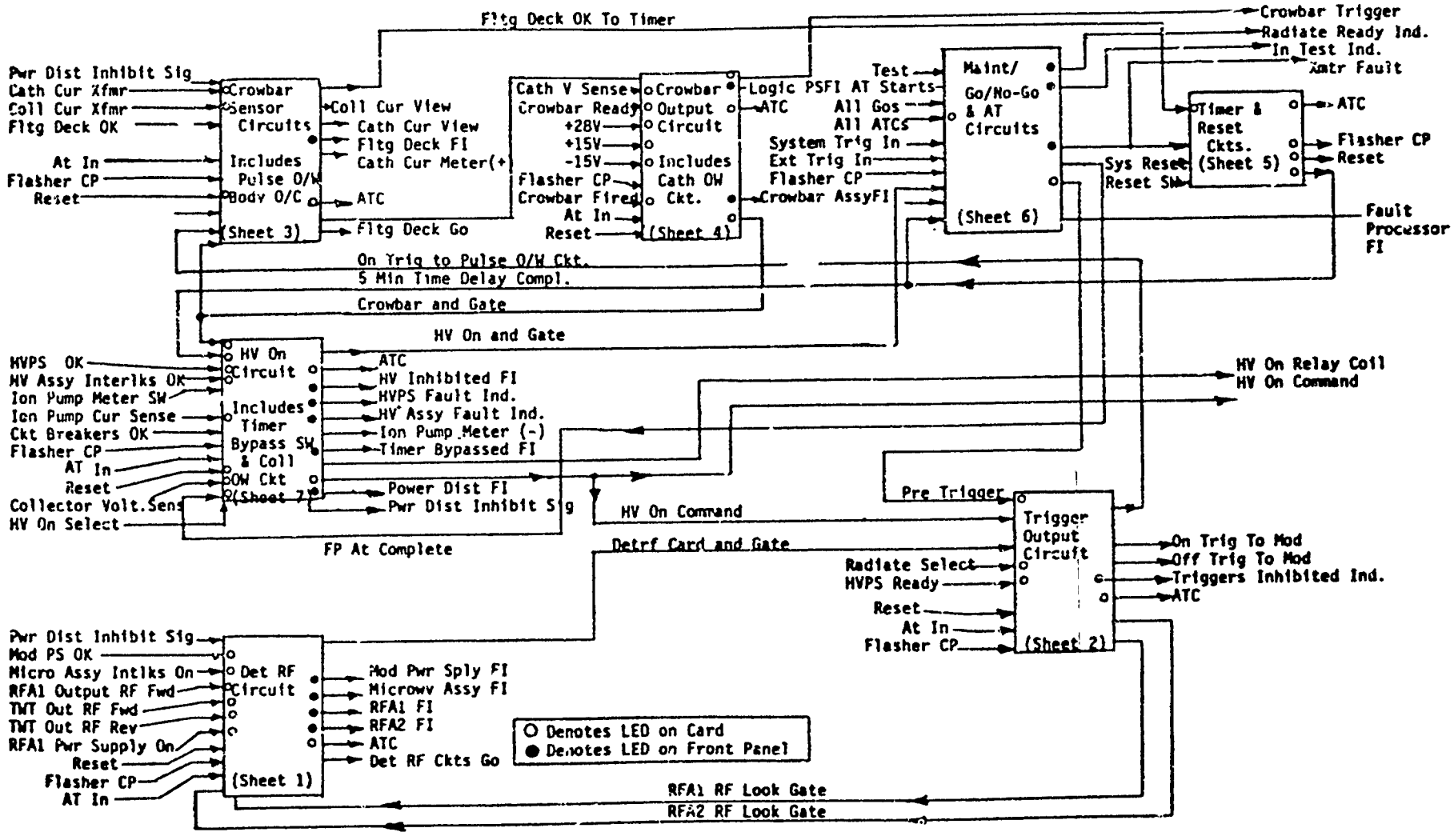



**FIGURE 6-2 BIT INDICATOR/SIGNAL FLOW DIAGRAM.**

## SECTION 7

## DISPLAY & EVALUATION OF BIT

**7.1 Fault Indicators:** Fault-detection devices sen fault signals and provide visual or audible indications of the condition of the equipment being monitored. They can be employed at the system, subsystem, equipment or printed-circuit board level to pinpoint malfunctions. In many cases, these indicators provide a continuous monitoring of system performance during operational missions. Fault indicators for BIT consist of displays, alarms and controls. Displays and alarms consist of:

a. Light-emitting diodes (LED),
b. Incandescent lamps,
c. Cathode-ray tubes,
d. Printers,
e. Neon lamps,
f. Liquid crystals,
g. Flashers,
h. Audio indicators, and
i. Magnetic latches.

Incandescent and neon lamps offer low-cost advantages and wide color range, but consume larger amounts of power and respond in the millisecond range. These indicators are used only where there is no restriction of power consumption and response time.

Audio indicators emit tones in the volume range of 75db, have life expectancies of 1,000 hours, and require 3 VDC at 35 MA or 12 VDC at 25 MA. These are more costly by a factor of ten over the LED. The size of magnetic-latching indicators is equivalent to audio indicators.

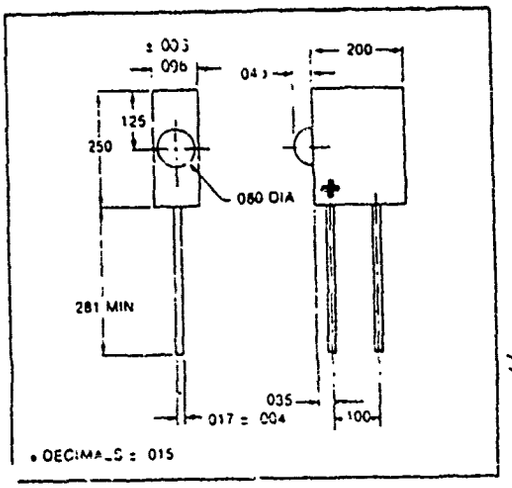
Solid-state flashers are available for applications requiring high visual alerts. They operate on 5 to 15 VDC at 10 MA and have useful lives greater than that of audio indicators, but less than those of LED.

Examples of packaged LEDs with other functions and audio alarms are shown in Figure 7-1.

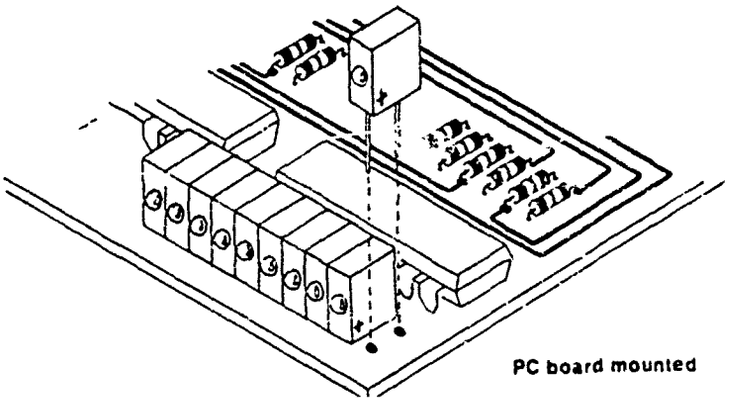
A typical card layout is shown in Figure 7-2 in which the BIT LEDs are located on the outside edge of the card shown at the top of the figure. These LEDs are available from multiple sources, in multiple colors, and in straight and right-angled versions packaged in cases similar to those used for CK05 ceramic capacitors. All of the designs are adaptable to easy loading on flow-soldered cards. LEDs, because of the nature of the materials from which they are made, are susceptible to soldering-heat problems. For this reason, cautionary words will have to be included on drawings advising assembly personnel not to apply heat for excessively long periods when soldering LEDs in the circuit.

Manufacturers guarantee LEDs for the life of the equipment on which they ·· installed or for ten years, whichever is less. Red LEDs, now in life testing, are expected to have a minimum life of ten million hours. Compared to the life expectancy of incandescent indicators, LED BIT indicators, even though they are constructed of plastic, are many orders of magnitude more reliable.

LEDs also permit smaller integrated-circuit driver-type logic to be used because they do not have the ten times normal surge currents experienced with incandescent lamps. In addition, since LEDs are mounted on cards, they are much more resistant to shock than are incandescent lamps.




actual size




Sample Latching BIT Indicator

Sample LED Indicator

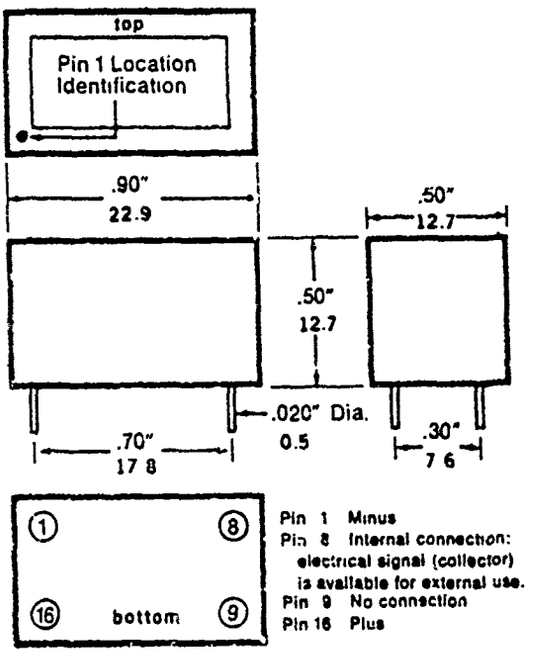
**Specifications**

| DC volts | Max. Current | Sound Output* | Temperature Range—°C | Weight |
|---|---|---|---|---|
| 3 | 35 mA | 76 dB | −50° to +50° | ⅛ oz 3.5 gm |
| 5 | 20 mA | 76 dB | −50° to +50° | ⅛ oz 3.5 gm |
| 12 | 25 mA | 76 dB | −50° to +50° | ⅛ oz 3.5 gm |

*measured at 12" Sound output varies with voltage.

**Application note:**
Electronic signal available at pin 8 (collector) can be used to feed an audio signal to a remote location; to an audio amplifier; or connected internally as a logic clock generator.

**Dimensions**




Sample Audio-Alarm Indicator

**FIGURE 7-1 EXAMPLES OF BIT INDICATORS.**

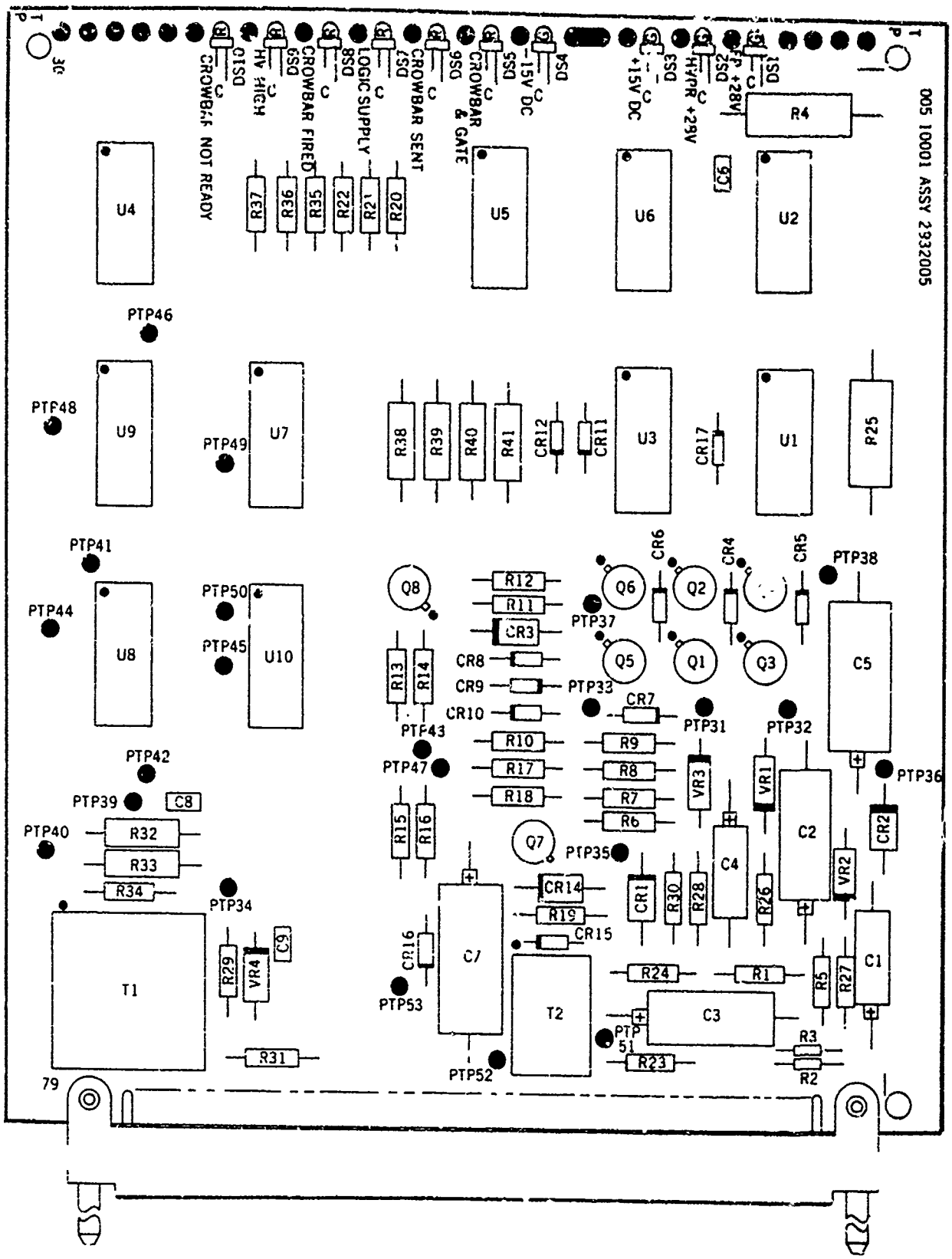



**FIGURE 7-2 TYPICAL CARD LAYOUT SHOWING BIT INDICATORS.**

**7.2 Centralized Fault Isolation:** Centralized active BIT can be used to provide local fault indication through LEDs, but it is likely to be more convenient to provide a computer driven printout at a central location. In addition, LEDs converted to individual passive BIT circuits can be located on a central fault status board rather than at the individual locations. Either way, a central display is presented which must be interpreted either by the computer or manually. If each fault signal represented a separate module there would be little problem. A more cost effective display might use sensors with greater coverage time single modules (or computer initiated tests with similar ambiguity in locating failures) and analyses of the indications of several sensors (or results of several test sections) to isolate to the failed modules. In general, this would be done by software, but an understanding of the process may be helped by considering how it can be done manually, as illustrated by the following occasion of the NELC FAST system.

**7.2.1. The Fast System:** Fault Analysis Systematic Tabulation (FAST) is a tabular approach to fault finding used to localize faults to a small set of replaceable modules. Using a process of elimination where the survivors are the failures, it is intended primarily for fault isolation in the field, but can also be used as an analytic tool during design to determine if the BIT and procedures are adequate. FAST is amenable to both manual and computer operation. The following example offers a brief description of FAST.

FAST is based upon a tabular presentation of system behavior under different input conditions or modes of operation. Table 7-1 has a column for each replaceable module and a row for each test performed. The entries define the modules involved in a given test and the roles they play. A dot in the upper right corner signifies a data path, while a dot in the lower left corner denotes a control function. An "X" in the box means that the module could be the cause of failure in the particular test. This must be determined by an engineer who possesses a detailed knowledge of system (or unit) operation. There may be some degree of subjectivity involved here since selection may be based on inferential reasoning.

Test is not confined to BIT, but is defined here as the observation of a particular response for a known set of input conditions, both control and data. The tests may encompass all modes of operation: normal, BIT, calibrate, off-line, and self-test. There may be several tests associated with a given set of input conditions. Thus, a test which requires the detection of display characters (whether good or bad) would be primarily indicative of defects in the data path, while a test which determines whether any characters are displayed at all (or whether they change) would indicate defects primarily in the control circuits.

Each test represents a row in the table and one can expect more than six defective candidates or entries per row. However, after several tests, the number of defective candidates will be reduced to the prescribed six or fewer. Further tests will reinforce and narrow the selection.

The construction of an effective table can be facilitated if certain design practices are observed and certain information formats are followed. Wherever possible, data paths should be allocated to modules which do not include control functions. This requirement can be satisfied, in most cases, since the MSI devices utilized in data paths (multiplexers, registers, serial/parallel converters) are not found in control functions (counters, gates). There is, of course, some overlap to be expected in gate and flip-flop modules.

Control circuits tend to thread through cascades of gates extending over more than six modules. ROM provides structured logic forms offering efficient implementation of combinational control functions.

To ensure field failures, unless catastrophic, will involve only one device or one module. BIT circuits should be separate from the circuits required for normal operation so that malfunctions of BIT modules will not effect normal system operation.

The unit block diagram should be sufficiently detailed so that each block function can be associated with a

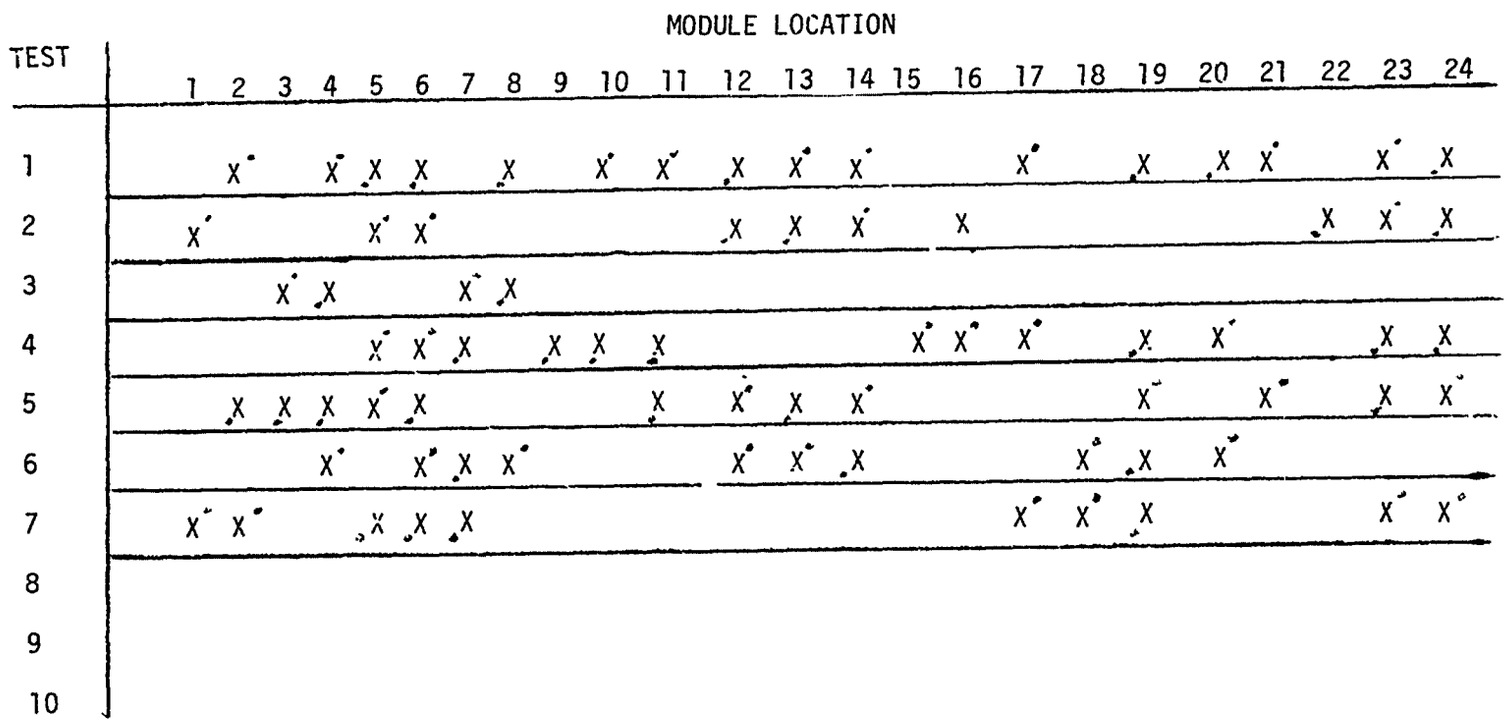



**TABLE 7-1 FAULT TABLE EXAMPLE**

set of cards. The card locations should be included within the block along with the identification of the detailed logic drawing. With the extensive use of MSI devices, the block functions will often correspond to device functions; there may be a 1-to-1 match. This format affords great visibility and traceability and can be used to develop the fault table without resort to the detailed logic drawings.

Fault table selection procedures can best be explained by using the example shown in Table 7-1. Here, for brevity, the tests are simply numbered. Most of the tests, if performed individually, cannot resolve the fault down to six modules. In fact, only test three has this ability.

Each of the tests is considered independent with respect to the order in which they are performed. The outcome of each test is a pass or fail. If test one fails, all the "Xs" of the first row are fault candidates and further testing is necessary. If test two fails the "Xs" common to rows one and two remain as candidates and these are locations (columns) 5, 6, 12, 13, 14, 23, 24. Test three should pass but it does not help in the localization. Test four, whether pass or fail, provides useful information. If test four fails, the overlapping entries reduce to locations 5, 6, 23, 24 and the fault-localization criterion is satisfied. If test four passes, several fault candidates ar removed (namely 5 and 6) and the criterion is again satisfied. Further testing can narrow the choice. Failure in test six would pinpoint location 6, a pass in 6 would remove location 6 from the fault list.

A note of caution must be injected. The entries do not indicate module utilization. For example, one data path may utilize part of a module while another data path uses another part. It may be necessary to denote, along with the "X", which section half of the module is used. This precaution is only necessary when a reduction based upon a passed test is attempted.

Fault-localization tables should be developed for each unit. The test conditions and pass-fail criteria should be listed. Manual procedures are very simple. If a test fails, draw vertical lines through all columns that do not have an "X" entries in that row. The "Xs" in these columns are eliminated from further consideration in any subsequent test. If a test passes, draw a horizontal line through the test row. Wherever an "X" appears, that column is then eliminated by drawing a vertical line through that column.

Consider the previous example at which test one fails. Draw lines down columns 1, 3, 7, 9, 15, 16, 18, 22. Next test two fails. Draw lines down columns 2, 4, 8, 10, 11, 17, 19, 20, 21. The remaining suspect columns are 5, 6, 12, 13, 14, 23, 24. Since the columns associated with test three have been eliminated, it is not necessary to perform this test. If test four passes, draw a horizontal line through row 4 and then vertical lines through columns 5, 6, thereby leaving only columns 12, 13, 14, 23, 24. If test four fails, draw vertical lines through columns 12, 13, 14 leaving columns 5, 6, 23, 24 suspect.

The tabular manipulations described above are all amenable to simple computer processes. The fault table is resident in memory and is addressable by test codes. In BIT modes, these codes are computer-generated and computer-tested. In other tests, the operator may have to request the test and enter the results.

The "X" entries of a table are binary coded with the bit positions in the computer word matching the module location. With a 16-bit computer word, three words would be necessary to represent a test row for a 2-row card baseplate. The correspondence would be as follows:

| Module Location | Computer Representation |
|---|---|
| 101 | Word 1 Bit 1 |
| 102 | Word 1 Bit 2 |
| 116 | Word 1 Bit 16 |
| 117 | Word 2 Bit 1 |
| 124 | Word 2 Bit 8 |
| 201 | Word 2 Bit 9 |
| 224 | Word 3 Bit 16 |

The programming effort can, of course be reduced by developing a common-row format to accommodate the 3-row (or 4-row) baseplate for both horizontal and vertical drawers. The fault-location algorithm is exceedingly simple. Perform a logical AND operation on all the test rows that fail; perform a logical OR operation on all the test rows that pass. The defective module is among those obtained by logically ANDing the results of previous AND operations with the complement of the results of the previous OR operations.

## SECTION 8

## COUPLING AND SHIELDING

Incorporation of BIT into a system poses problems of coupling and shielding. BIT must be designed so that it does not interfere with mission operations. This is more difficult with active BIT than with passive. However, both active and passive BIT can provide transmission paths for electromagnetic interference.

A significant complexity is added to the system when the testing mechanism requires isolation. Figure 8-1 shows a picture of the isolation required to allow a decentralized test pattern to test a module and not affect another module's activities. Not only is this isolation circuitry a significant overhead hardware requirement, but it places additional delays in circuit modules in the normal operational path flow.

**8.1 Digital Transmission of Analog BIT Data:** Another contribution of integrated-circuit technology is the miniaturized Analog-to-Digital Converter (ADC). Small size (postage-stamp size and up) and modest cost make it feasible to locate these miniature ADCs at remote sensor sites so that data can be transmitted in digital format.

The advantage of this approach is that problems of analog-signal phase shift, waveform distortion, and attenuation are eliminated. Filtering and amplification requirements are greatly simplified, and the problem of noise interference is reduced appreciably.

Some of the most bothersome problems of data transmission over lines of any appreciable length have always been those of interference due to electrical currents circulating in a ground loop and the difference in ground potentials at the source and receiver ends of the transmission line.

While any application of BIT has a potential for electromagnetic interference problems, BIT of transmitters is perhaps the most difficult application. Because of the many different noise levels encountered in a transmitter, a wide variation of coupling and isolation techniques may be employed according to the noise and voltage levels which are encountered. Four methods of signal coupling are shown in Figure 8-2.

**8.2 Line Drivers And Line Receivers:** The most common method of coupling BIT signals between units, in which less than 10 to 15 volts of common-mode noise immunity is anticipated, is the line-driver/line-receiver combination shown at the top of Figure 8-2. Several line-driver units are available to the BIT circuit designer. They typically operate in conjunction with 5 volt logic from a positive volt supply. A typical pair is the 9614 line driver and 9615 line-receiver, for transmitting digital data in a noisy environment when there is a 15V differential in the equipment grounds.

**8.3 Transformer Coupling:** Most currents which are monitored in a transmitter originate in high-voltage areas subject to extremely high common-mode signals. If a line-driver was employed with several feet of wire, it could easily pick up a few hundred volts of common-mode signals and destroy the units. Here then, is a place where a well-shielded current transformer and a circuit such as that shown in the upper part of Figure 8-2 are used. Current transformers typically have rise times of 10 ns so relatively swift reaction time can be attained at a low output impedance. Fifty to 100 ohms is common. This BIT signal can be fed directly into a comparator-type integrated circuit or through a balanced transmission line to the primary of a center-tapped, shielded, isolation-coupling transformer. One end of the secondary of the coupling transformer is grounded and the other is fed into a comparator circuit where the signal is amplified up to the level required by the digital logic circuitry.

**8.4 Coupling Through Line Filters:** A third method of coupling, especially useful on slow signals,

such as those from flow or pressure switches, is also shown in Figure 8-2. Here, a signal can be coupled from a noisy environment to a clean logic area directly through a miniature EMI filter. This is commonly known as a "brute-force" method. However, since no active components are used, it is extremely reliable in a moderately noisy area.

**8.5 Optical Coupling:** The last method shown at the bottom of Figure 8-2 is optical coupling, which is useful where a difference in voltage, as high as many kilovolts, exists between the level of the sending unit and that of the receiving unit. This method can respond to signals in the tens-of-nanoseconds range and can be designed to be extremely noise immune when the sending and receiving units are properly shielded. These optical devices are available from several manufacturers. Various speeds are available with rise times of about 10 microseconds down to several nanoseconds. Generally, the faster the devices, the smaller are the signals coming from the output of the coupler.

Fiber optics usually represent the safest method of coupling signals across a high-voltage interface. Fiber optics are available in two materials, plastic and glass. Fiber optics of glass, typically, have a higher temperature range and are generally more compatible with the environments, including temperatures, encountered in military transmitters.

Where only a few hundred volts of isolation are required, integral units known as optical couplers are available which control both the sending and receiving photodiodes or photo transistors. These units are especially valuable for coupling BIT signals in transformerless power supplies or magnetic modulators, where all busses and circuits are floating with respect to ground, but command signals arrive referenced with respect to ground.

The most serious objection to using an optical transmission line has been the difficulty of repairing a broken line. However, this problem has been overcome and most manufacturers have developed techniques to repair broken optical lines. The same techniques can afford a means for routing the signal to alternate destinations without extensive recabling.

**8.6 Dynamic Range Of Noise Level:** BIT levels for both pulse and DC monitoring circuits generally are confined to the levels between zero and ± 28 volts, maximum. This facilitates the use of miniature capacitors and filters. Because of the possibility of a difference of DC or AC potentials existing between chassis grounds of the sending and receiving circuit, effective use of small signals in the millivolt range in the vicinity of a transmitter becomes rather questionable and is usually costly to implement. Therefore, reliable circuits should normally be designed to operate with signals in the area from 5 to 12 volts. This produces a circuit with a reasonable size signal for use in transmitter noise environments. Circuits which interface with normal, external, on-line automatic test equipment typically must have signals within the range of zero to ± 10 volts and, again, signals in the millivolt range and up to 0.65 volts should be avoided. In transmitters especially, good shielding and proper grounding determine the BIT fault indicator accuracy to the greatest extent. In extremely high-power transmitters, magnetic shielding of the fault logic plays an increasingly important role in the BIT indicator accuracy.

The most difficult (but probably the most important) time in which transmitter BIT indicators are needed to function properly, is during high-voltage breakdown or during times in which the high-voltage power supply is deliberately shorted, as during a crowbar. During these intervals, much higher than normal currents are present and extremely high magnetic fields occur. To achieve proper fault indicator accuracy at this time, magnetic shielding must be used to reduce magnetic field effects in areas where logic is located.

Good shielding costs money, but good shielding results in superior BIT indicator operation. If at all possible, it is best to shield against magnetic fields near their source. However, most often in military

equipment, weight is also an important consideration. Therefore, magnetic shielding for only BIT and fault circuitry may become more desirable. Cold-rolled steel is a good economical compromise as a means for magnetic shielding. Magnetic shielding materials, which are prone to take on a permanent bias, should be avoided, as their initial good shielding qualities may soon be lost after only a small number of arcs.

Examples of the rewards of good magnetic shielding are in two linear accelerators located at Stanford University. In the small accelerator, magnetic shielding of the high-power modulators was minimal. In the larger accelerator, much care was given to get good continuous magnetic shielding around the high-power areas. The control and fault circuitry is essentially the same for both. However, the larger unit with its higher power and added complexity has been proven orders of magnitude more reliable, primarily due to the superior magnetic shielding.

Proper choice of the protective-logic family with good degree of noise immunity is also beneficial to achieve reliable failure indicator accuracy. There are several families of logic available. Typical noise immunities and supply voltages are shown in Table 8-1.

| Logic Family | Typical Noise Immunities | Typical VCC |
|---|---|---|
| Resistor-Transistor Logic (RTLO) | 300mv | 3.6 |
| Diode Transistor, Transistor-Transistor Logic (DTL/TTL) | 800 mv | 5.0 |
| High Noise Immunity Logic (HNIL) | 5.0v | 15.0 |
| High Threshold Logic (HTL) | 5.0v | 12.0 |
| Complementary Metal on Silicon (CMOS) | 5.0v | 15.0 |

**TABLE 8-1 — LOGIC NOISE IMMUNITIES AND SUPPLY VOLTAGE**

It immediately becomes apparent from the noise immunity figures that HNIL, HTL and CMOS are the most advantageous logic families when only noise immunity (and not speed) is the main criterion. Looking further into the logic families, HNIL has good voltage-noise and power-noise immunity. HTL tends to be essentially the same, but CMOS, with essentially the same voltage-noise immunity, has a higher input impedance at the input on its gates, even though the output impedances are comparable with HNIL and HTL. During nearby arcing. CMOS can cause a greater degree of false indications under identical magnetic- or capacitive-induced transient conditions due to the transmission-line effect described in this section. So, where only noise immunity is concerned and where neither speed nor power-consumption considerations play an important part, HNIL is somewhat superior.

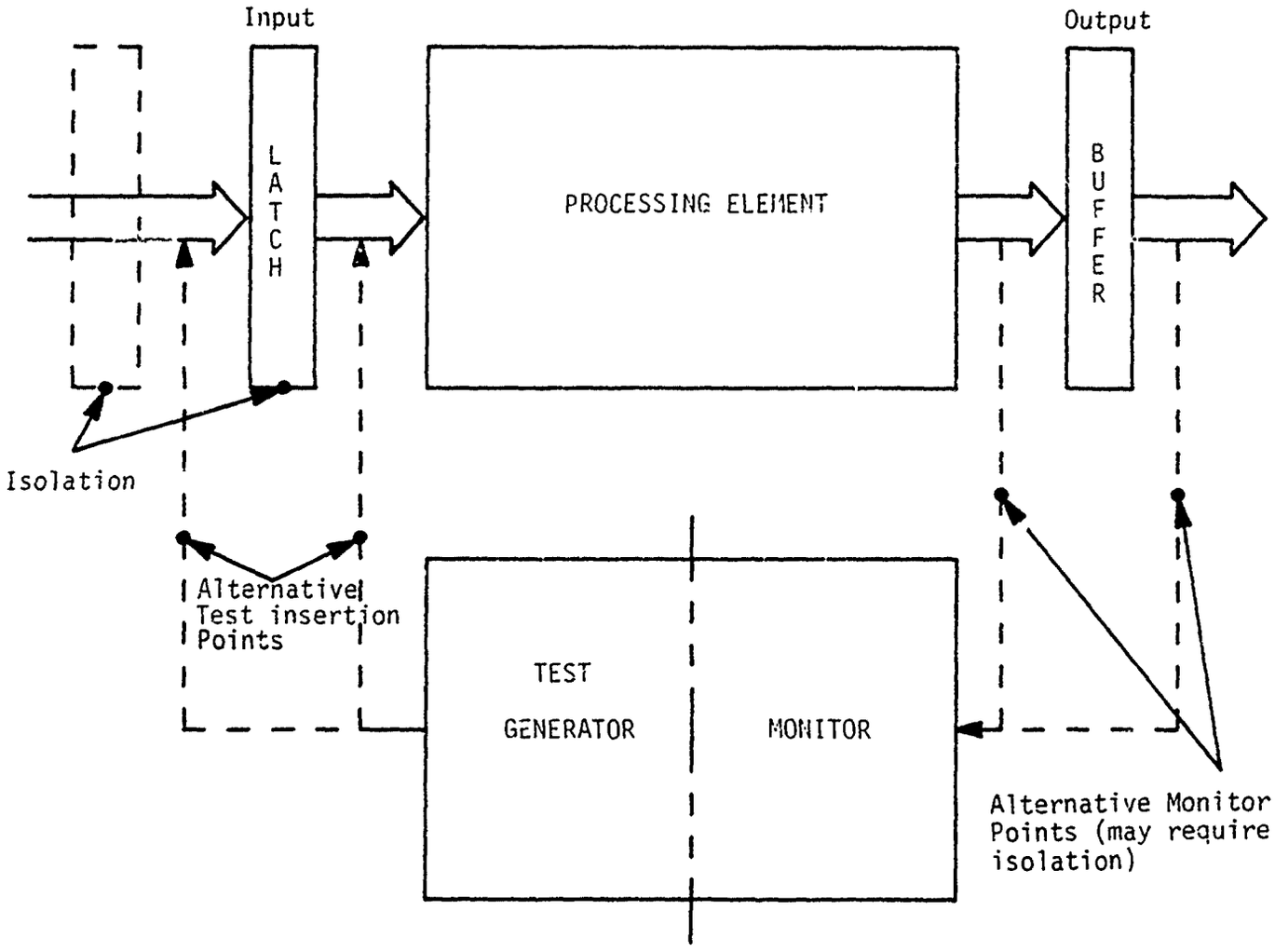



**FIGURE 8-1 DECENTRALIZED TESTING ISOLATION REQUIREMENT**

# SECTION 9

# ANALYZING AND OPTIMIZING BIT

The preceding sections discuss the many factors that the BIT designer must consider in his design. This chapter provides some of the analytical techniques he can use to evaluate and optimize his design. The methods are compiled from several sources and hence some different approaches to the same consideration will be noticed. In addition, the purposes of the techniques differ. For example, some are concerned with evaluating the effectiveness of a particular unit circuit while others are concerned with optimizing the location of test points. The designer should review these techniques and use these techniques, combination of techniques, or modifications which meet his needs.

## 9.1 Evaluation of Passive BIT

### 9.1.1 Basic Assumptions

The BIT circuitry envisioned here is anticipated to exist as part of a family of functional digital modules. It is intended to monitor and validate module operations and provide a go/no-go indication. The level of diagnosis in this situation is synonymous with the detection level. Off-line test insertion capability is not considered in this suggestion.

Module performance validation will focus on the detection of functional faults as opposed to electrical or timing faults. This basically identifies logic faults (gate behavior changes) which alter the output from a desired value. Transients will not be filtered at the module level. If a transient causes an error which is detected, then the no-go indication will be given.

### 9.1.2 Goals

For the type of monitoring circuits being considered, the designer must be able to evaluate both the cost and effectiveness of various BIT schemes. The cost measures can involve time, space. power, and reliability.

An important aspect of the analysis comes in trying to quantify the fault detection performance of a particular BIT approach. The primary goal of Section 8.1.4 is to define the measures used to evaluate the effectiveness of the alternating BIT techniques.

Section 8.1.3 suggests other performance measures that give additional information about the completeness of the BIT fault monitoring. These measures are included because it is apparent that the present performance criteria are not totally adequate for determining the overall effectiveness of on-line, concurrent fault monitoring techniques.

### 9.1.3 Cost Criteria

If a broad interpretation is applied to cost, then there are a significant number of parameters which might be considered. This report focuses on cost parameters in four basic areas, 1) time, 2) number of packages, 3) power consumption, and 4) reliability. Each parameter in each of the basic areas will be defined in the following subsections.

#### 9.1.3.1 Time

The time required to apply an n-cycle off-line test is a measure of the amount of time the module is used for

self-testing and is not avilable to do normal processing. This parameter is defined as follows:

> Number of Cycles for Test = The number of clock periods necessary to perform an off-line module self test.

The actual cost in lost processing time is dependent on the specific application,since in many systems there are times when a module is idle. This time can be used for module self-testing with no reduction in overall processing capability. However, many BIT techniques do not involve an off-line test method, therefore the number of cycles for test will be zero.

### 9.1.3.2 Number of Packages

The number of integrated circuit packages required to implement the recommended BIT approach must be determined. The basic idea of this parameter is to describe the portion of the module that is identifiable as BIT circuitry. This measure, expressed as a percentage is defined as

$$\text{Ratio of BIT Packages to Total Module Packages} = \frac{100.\ [\text{Number of Packages in BIT}]}{\text{Total Number of Packages (including BIT)}}$$

The strongest tie to actual cost in terms of dollars and space is given by package count. The final cost of a board assembly is usually less sensitive to the cost of a package than it is to the cost of placing the package on the board. Size is also somewhat more related to packages than gates or other circuit complexity measures.

One can develop a scaled package count with the pins per package as a scale factor. This approach holds board space as the most important aspect of this measure, although larger packages do tend to cost more. While this is a defensible argument, it seems that, for the added difficulty, little additional information is gained.

### 9.1.3.3 Power Consumption

The determination of the power consumption of the BIT circuitry can be treated in a straightforward manner. Even at the functional design stage, a fair estimate may be made as to the explicit packages required for a particular BIT approach. Once this is done, the typical DC power requirements for the required packages may be summed to obtain the total BIT power requirements. This leads to the following definition

$$\text{Power Consumption of BIT} = \sum_i PB_i$$

where the sum is taken over all of the packages in BIT and PB i is the typical power consumption (product of typical current and typical voltage supply) for the i-th BIT package.

Another cost measure was defined to give an indication of the proportion of the power consumption of BIT relative to the whole module. This parameter, expressed as a percentage, is defined below

$$\text{Ratio of BIT Power Consumption to Total Module Power Consumption} = \frac{100 \quad \sum_i PB_i}{\sum_j PM_j + \sum_i PB_i}$$

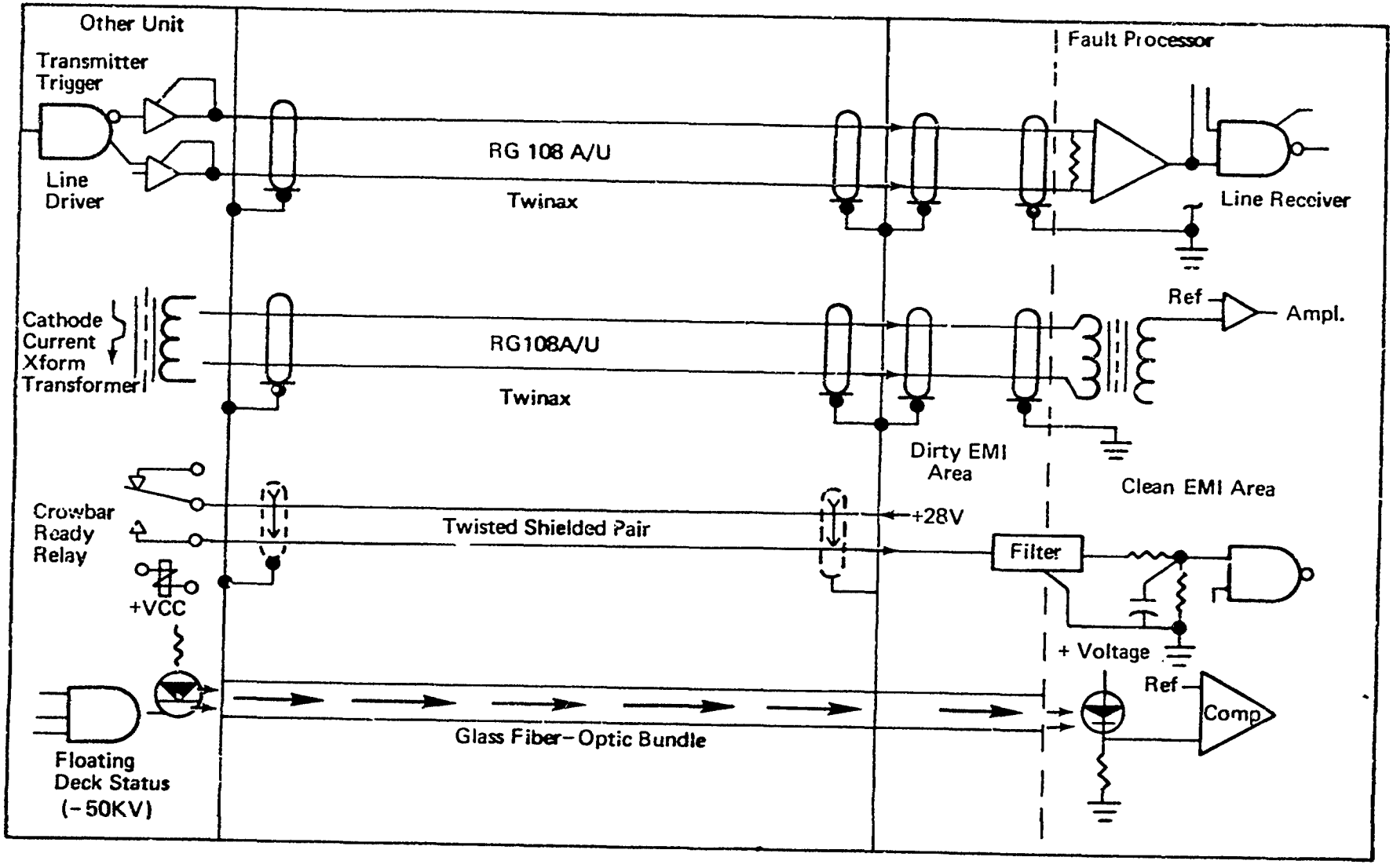



**Figure 8.2 Typical Transmitter BIT Interface Methods.**

where PB i is the typical power consumption of the i-th BIT package and $\sum$ j is the sum over all the non-BIT module packages.

### 9.1.3.4 Failure Rate

The addition of hardware to affect fault detection without repair adversely affects the reliability (MTBF and failure rate) of the module. This is tolerable only if the added circuitry improves the repair time and hence system availability. The purpose of this measure is to indicate the impact of the BIT on the MTBF or failure rate. It is much more difficult to qualify, without extensive system operational field data, the resultant mean-time to repair (MTTR) improvement.

The failure rates (FR) for individual packages found in modules are calculated using MIL-HDBK-217B.

The failure rate of the BIT circuit is the sum of the failure rates of the BIT packages. The measures used to express the failure rate information are defined as

$$\text{Failure Rate without BIT} = \sum_j \text{FRM}_j, \tag{3.5}$$

$$\text{Failure Rate with BIT} = \sum_j \text{FRM}_j + \sum \text{FRB}_i, \tag{3.6}$$

where $\text{FRM}_j$ is the failure rate of the j-th module package or discrete component (non-BIT) and $\sum_j$ is the sum over all of the non-BIT packages and discrete components. $\text{FRB}_i$ is the failure rate of the i-th BIT package or component, and $\sum_i$ is the sum over all of the BIT packages and components.

To indicate explicitly the relative proportion of the BIT failure rate to the failure rate of the whole module, a new cost measure was defined. This parameter, expressed as a percentage, is defined below

$$\text{Ratio of BIT FR to Total Module FR} = \frac{100 \cdot \sum_i \text{FRB}_i}{\sum_j \text{FRM}_j + \sum_i} \qquad \text{FRB}_i$$

where the variables are defined as above. The mean-time between failure (MTBF) is related to the failure rate by

$$\text{MTBF} = \frac{1}{\text{FR}} \tag{3.8}$$

### 9.1.4 Performance Criteria

This subsection is concerned primarily with understanding how "good" a job a particular BIT approach does. Clearly a sound quantitative measure is required if reasoned decisions are to be made. The nature of complex digital systems testing is such that something more than intuition must direct decision making. The focus of this section will be on BIT with a passive monitoring function only. However, many of the ideas and results can readily be applied to the evaluation of active BIT capable of input control (test word insertion).

The two performance measures used in this report are percent of packages monitored and percent of gates monitored. These measures generally correspond to two other measures sometimes used, namely, percent of function tested and confidence level, respectively. The measures used in this report are intended to have names that suggest their definitions and to be defined more explicitly than previously used measures.

### 9.1.4.1 Percent of Packages Monitored

Fault monitor coverage can be related to the percent of the total module packages monitored. This performance measure is defined as

$$\text{Percent of Packages Monitored} = \frac{100 \cdot \text{Number of Packages Monitored (Including BIT)}}{\text{Total Number of Packages (Including BIT)}} \quad (3.9)$$

A package is said to be monitored if faults within that package that cause an erroneous result cause the pass/fail output signal to indicate a fault. In practice, this definition is relaxed somewhat to include all those packages in an array whose "data" outputs are fully monitored, but have a small number of "status" outputs which are not verified. (Example: a string of counters whose final ripple carry output is not checked.)

### 9.1.4.2 Percent of Gates Monitored

Because of the wide range of integrated circuit complexities within the TTL family used to implement the QED modules, a more realistic indication of the percentage of total possible faults that can be detected may be given by the measure defined below.

$$\text{Percent of Gates Monitored} = \frac{100 \cdot \sum_i G_i}{\sum_j G_j} \quad (3.10)$$

where $G_i$ is the number of equivalent gates in the i-th package, and $\sum_i$ is the sum over all of the monitored packages (including BIT). $\sum_j$ is the sum over all of the module packages (including BIT). A package is defined as monitored in the same manner described above. In determining the percent of gates monitored it is possible and even desirable to look at the individual gate level diagrams of the partially monitored packages, decide how many gates are indeed monitored with each BIT circuit and add these numbers to the numerator of the definition. This approach was not taken because the additional accuracy is not essential to the objectives of this study.

### 9.1.4.3 Other Possible Performance Measures

Since the monitoring circuit considered here is passive, it cannot initiate a test for the presence of a given fault. Therefore, it seems reasonable to only attempt to measure how effectively it will monitor and validate module results given the data inputs which occur. The name monitoring capability index (MCI) is chosen for the measure to be defined subsequently. It is important to understand that a high monitoring capability index and a valid pass/fail line indicate proper module operation only for that portion of the module that the input word set has exercised. Under these conditions the user can be confident that the output data is valid, but the user should not assume that the module is fault-free. The MCI indicates the potential to detect failures given the input conditions necessary to produce detectable errors.

The level of monitoring capability depends upon three things:

(1) The percent of the function monitored (function coverage);

(2) The percent of the module fault conditions which can be detected given proper inputs (fault coverage), for the monitored portion of the circuit, and,

(3) The percent of time that the results are monitored (cycle coverage).

Each of these three properties is now described in more detail.

### 9.1.4.3.1 Function Coverage

A package is said to be *covered* if all of its inputs or predecessors of its inputs are monitored and its outputs or successors of its outputs can be monitored and verified.

In order to develop some scale of monitored versus unmonitored behavior it is necessary to count packages, gates or another quantity. In fact, the number of faults which fall within the monitored area would be of highest interest. It is often difficult to provide this number. A reasonable compromise is to use failure rate as a scale of fault presence and to define the *percent function monitored* (PFM) as

$$\text{PFM} = \frac{\text{FR of covered packs} \cdot 100}{\text{FR of the Module}}$$

Both of the FR numbers include the BIT circuitry itself.

### 9.1.4.3.2 Fault Coverage

An example of complete fault coverage is given in Figure 8-1 and partial input coverage which may result in partial fault coverage is shown in Figure 8-2. The BIT of Figure 8-1 does all that the operational module does for every input combination and this will detect all faults assuming no failure in the BIT. The approach demonstrated in Figure 8-2 is motivated by its flexibility and reduced expense. However, only a limited number of input cases are checked by the monitor. Hence, presumably less than full fault detection results. A parity checker is an example of 100 percent input coverage but less than 100 percent coverage.

In the most detailed view of this parameter, the fault coverage is more dependent upon coverage of input values which check a large number of faults than inputs which do not check many faults. That is to say the index is a measure of the number of faults which can be monitored with respect to the total number of faults. A weaker measure, but one which is easier to obtain, is to count the number of input cases which can be monitored versus the total number of input cases. In general, the number of input patterns required for a complete test will be smaller than the total number of input patters but harder to determine. These two points of view are summarized by the following definitions:

$$\text{Percent Input Coverage (PIC)} = \frac{100 \text{ Number of Inputs, I, for which f(I) can be Verified}}{\text{Total Number of Inputs for which f(I) is Defined}}$$

and

$$\text{Percent Fault Coverage (PFC)} = \frac{100 \text{ Number of Faults which can be Detected}}{\text{Total Number of Faults in f(I)}}$$

(For another approach see Section 8.3)

### 9.1.4.3.3 Cycle Coverage

For a module which monitors values continuously in time (as was the case in Figures 9-1 and 9-2) the present cycle coverage in 100 percent. When a sampling scheme is used (Figure 9-3), then there will be results which could be verified (with respect to fault coverage) but will not be because they are not sampled.

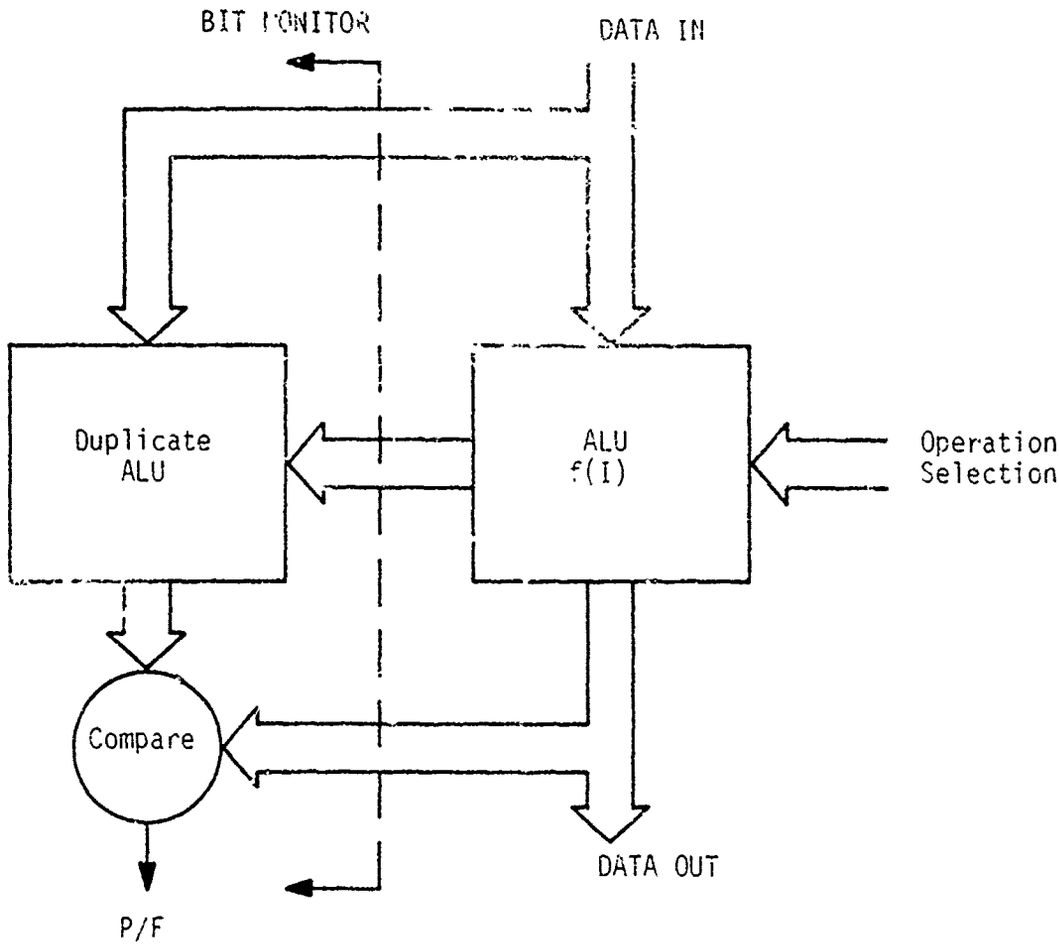



**FIGURE 9-1 MONITOR WITH 100 PERCENT COVERAGE**

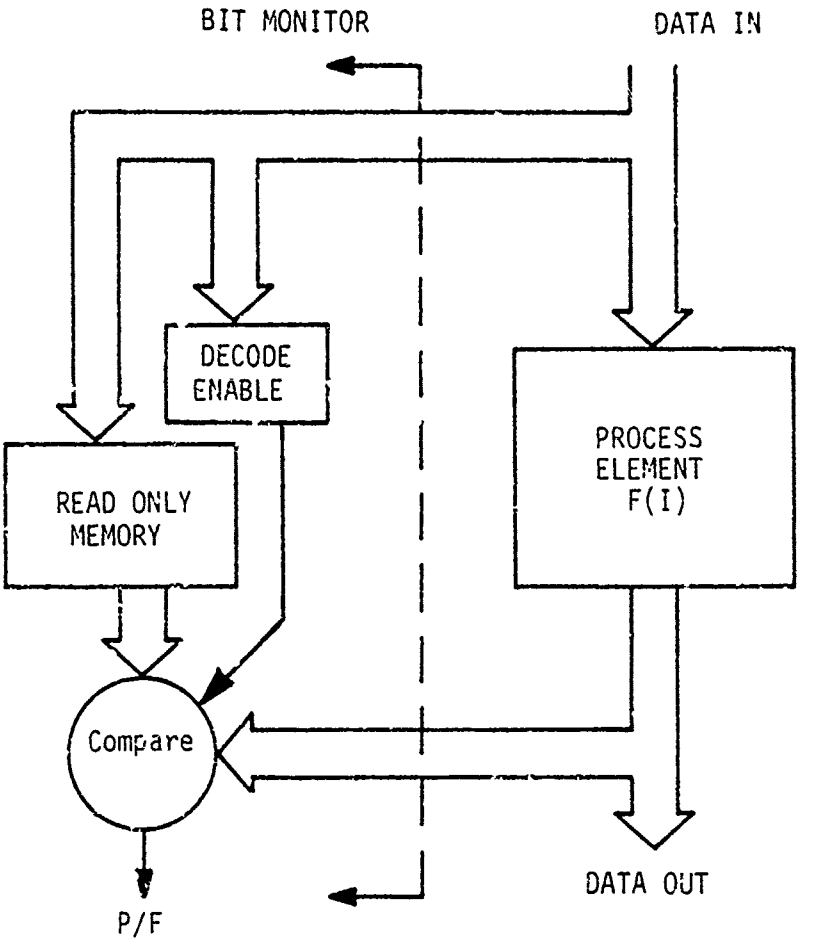



**FIGURE 9-2 MONITOR WITH PARTIAL INPUT COVERAGE**

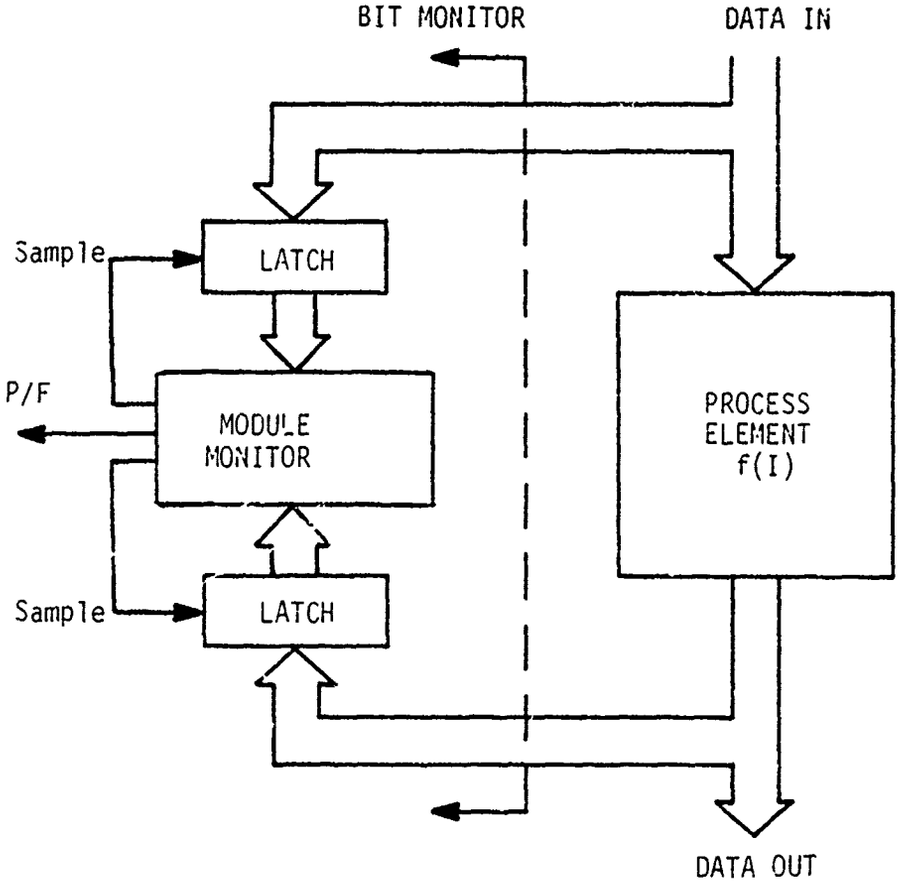



**FIGURE 9-3 BIT MONITOR WITH PARTIAL TIME COVERAGE**

The primary motivation in consi ing a sampling scheme is that by reducing the time demand on the monitor, it is possible to consider using a programmable device. Programmable devices are typically slower but more flexible and thus applicable to a wider range of monitoring tasks.

If the monitoring device has a sampling rate of S samples per second and the device has an operating rate of C cycles per second. then the percent cycle coverage is given by

$$\text{Percent Cycle Coverage (PCC)} = \frac{100 \cdot S}{C}$$

When the device being monitored is asynchronous, then the cycles per second measure may be taken as the maximum number of cycles possible for the device propagation delay.

For the second example (Figure 9-3) the entire module is monitored resulting in a PFM of 100 percent. The other measures are taken as hypothetical. Sampling hardware can, in general, readily provide 100 percent input coverage and must produce less than 100 percent cycle coverage. An example of cycle coverage can be given by typifying the sampling device as a microprocessor. Estimates of performance are given in Table 9-1.

| | | |
|---|---|---|
| Estimate of the Number of CPU Cycles to run an ALU emulation | = | 600 |
| Cycle time (Z-80 example) | = | 120ns |
| Total Emulation time | = | 72μs |
| S = Sample rate | = | $14.10^3$ /s |
| Worst case (fastest) delay of ALU | = | 110ns |
| C = Worst case (most) cycles/s | = | $9.10^6$ c/s |
| Cycle Coverage = 100 · S/C | = | 0.15% |

**TABLE 9-1 EXAMPLE CYCLE COVERAGE**

#### 9.1.4.3.4 Composite Measure

At this point it would be useful to attempt to define the overall idea of module monitoring capability as a composite of PFM, PIC, and PCC. For example one might consider a weighted sum of these components such that,

$$MCI = \alpha\ PFM + \beta\ PIC + \gamma PCC.$$

The problem then becomes one of choosing appropriate values for the coefficients $\alpha$, $\beta$ and $\gamma$. In such a formulation the weights are directly determined by the relative importance of the percent of the function monitored, the percent input coverage and the percentage of the total number of cycles covered. Additional work needs to be done to both validate this basic equation and to determine $\alpha$, $\beta$ and $\gamma$ or their equivalents.

As an example of an application of a composite measures, consider the following values:

PFM = 84%,
PIC = 100%,
PCC = 100%.

One might consider that the composite monitoring capability provided is given by an index of 0.95 (i.e., $\alpha = \beta = \gamma = \frac{1}{3}$). The confidence level,

For the ALU then

MCI: PFC = 100%
PIC = 100%
PCC = 0.15%

Intuitively, the cycle coverage seems less significant since the same amount of testing may be obtained by allowing more elapsed time. As long as the elapsed time does not become significant with respect to system time constants, the cycle coverage reduction may not practically affect the usefulness of the BIT.

Since considerations such as this are so dependent on system factors, it may be found that the composite measure can best be defined in a system dependent way.

In conclusion, if meaningful statements can be made about the data seen by a module in a certain period of time, then something can be said about the confidence level. For example, if a module with BIT which has PFM = 100%, PIC = 100%, PCC = 100%, and for which input data constituting a complete test set has been processed (within certain time limits), then the pass/fail indication may be interpreted with 100 percent confidence. Remove any of the above qualifiers and it is not clear what can be said about confidence level. Confidence level is generally a measure of active, not passive behavior. Therefore, if a BIT circuit has an active mode in which it is possible to insert test words, only then would it be possible to define a meaningful confidence level.

**9.2 Test Point Selection:** One of the key requirements for implementing a cost effective BIT is proper test point selection. The sole purpose of integral sensors is to facilitate performance monitoring and fault isolation. It is therefore obvious that the location of these sensors (test point selection) must be based on the ability of the sensor(s) to detect and isolate faults in the prime equipment.

Two additional factors must also be considered during test point selection. First, test point selection should be biased so as to concentrate sensors in areas where failures are most likely to occur. A key element in test point selection is therefore the establishment of probability-of-failure data for each functional area in the system. It is important to note that the failure rates are used only in a relative sense to bias test point selection to the areas in the system that are more likely to fail.

Secondly, there is the question of how feasible a sensor is once a test point is selected. If the parameter at a particular test point is extremely difficult to measure, that test point should be avoided in the interest of a cost effective system even if two alternate test points are required. This should generally not be a problem since most parameters can be sensed with cost-effective sensors. It is also important to note that test point selection should not be biased toward selection of specific test points to permit the use of existing or easily implemented sensors. The effectiveness of the BIT could very quickly be compromised by such an approach. The remainder of this section provides a detailed description of techniques used in a systematic test point selection procedure.

**9.2.1 Failure Rate Establishment:** Figure 9-4 shows a functional block diagram of a system showing the interrelationships between the units. The system consists of the IF processing portions of a pulsed radar channel, including those items necessary to generate test signals. Included are a Variable Gain Amplifier (VGA) with a Sensitivity Time Control (STC), a phase detector, an A/D converter, a digital integrator, a D/A converter, and a threshold circuit driving the display scope. A VCO supplies the system IF signal, clock and COHO. The synchronizer controls the timing for the system, generating the pulsed IF, the STC timing and the A/D sample signals. For the sake of simplicity, the power supplies needed for these units are not included in the diagram although it can readily be seen that the supplies could be LRU's providing inputs on which the other units would be dependent. Each LRU can be specified according to its inputs and outputs as follows:

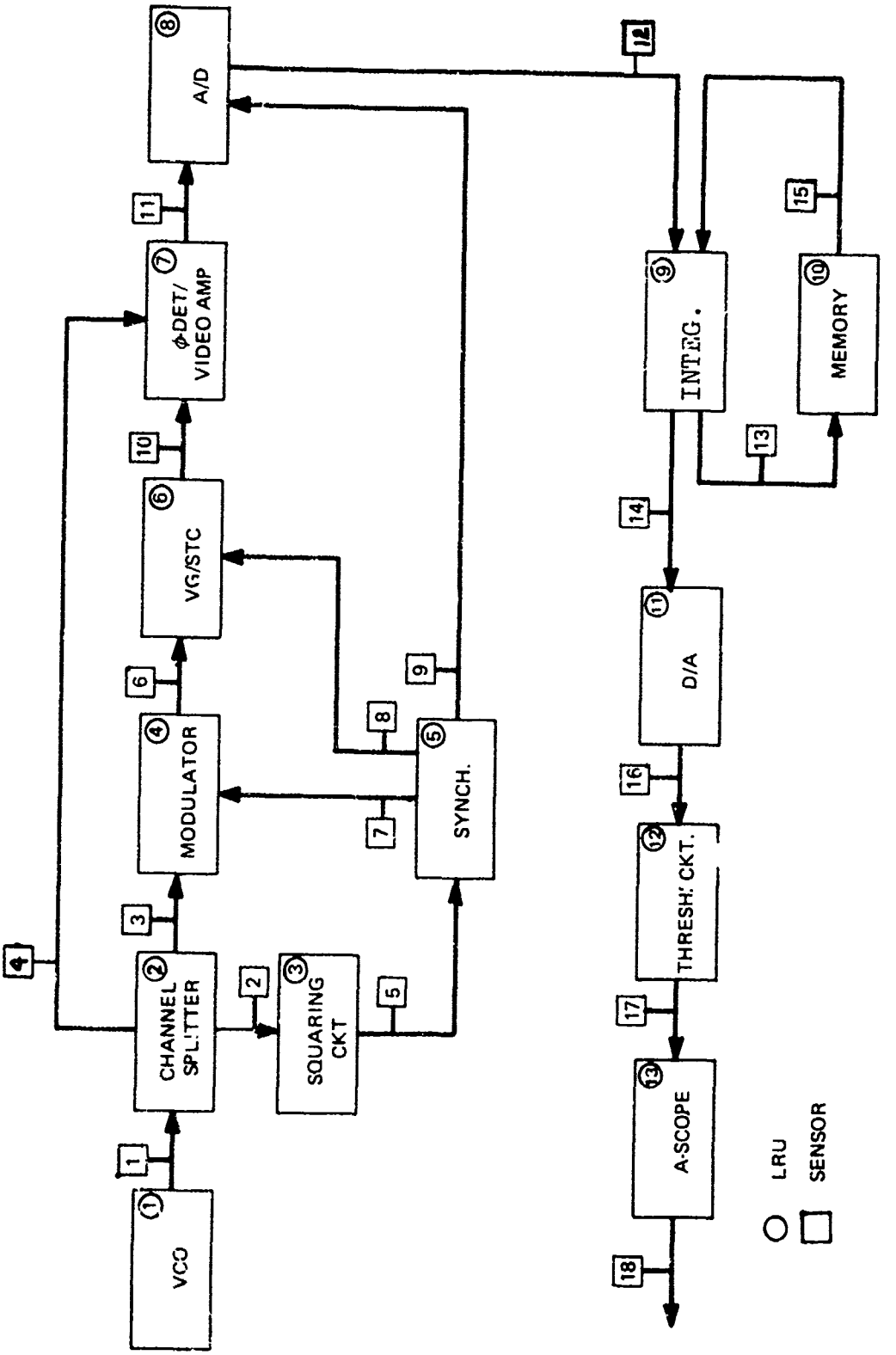



**FIGURE 9-4. EXAMPLE OF PRIME EQUIPMENT**

| LRU | Name | Inputs (from) | Outputs (to) |
|---|---|---|---|
| 1 | VCO | — | 2 |
| 2 | Channel Splitter | 1 | 3,4,7 |
| 3 | Squaring Circuit | 2 | 5 |
| 4 | Modulator | 2,5 | 6 |
| 5 | Synchronizer | 3 | 4.6,8 |
| 6 | VGA, STC | 4,5 | 7 |
| 7 | $\theta$ Det. Video Amplifier | 2,6 | 8 |
| 8 | A/D | 5,7 | 9 |
| 9 | Integrator | 8,10 | 10,11 |
| 10 | Memory | 9 | 9 |
| 11 | D/A | 9 | 12 |
| 12 | Threshold | 11 | 13 |
| 13 | A-Scope | 12 | — |

A particular LRU can be defined as having failed if it does not provide a valid output in the presence of its valid inputs. By using established reliability data (e.g. MIL-HDBK-217B) and knowing the LRU internal electronics, the likelihood of the failure of a given LRU output can be expressed mathematically. The normalized failure rate of those portions of an LRU concerned with relating a given output to the LRU inputs can be taken as the probability that this output will fail. Thus, the system can be described by a listing of the LRU's, the outputs, and their relation to the inputs. In the above example the numbers might be:

| LRU # | Output to LRU # | Failure Rate $\times 10^{-6}$ | Input |
|---|---|---|---|
| 1 | 2 | 13.1 | — |
| 2 | 3 | 4.3 | 1 |
| | 4 | 4.3 | 1 |
| | 7 | 4.3 | 1 |
| 3 | 5 | 10.1 | 2 |
| 4 | 6 | 12.6 | 2,5 |
| 5 | 4 | 16.7 | 3 |
| | 6 | 18.0 | 3 |
| | 8 | 21.3 | 3 |
| 6 | 7 | 15.3 | 4,5 |
| 7 | 8 | 12.0 | 2,6 |
| 8 | 9 | 25.2 | 5,7 |
| 9 | 10,11 | 11.3 | 8,10 |
| 10 | 9 | 15.6 | 9 |
| 11 | 12 | 20.1 | 9 |
| 12 | 13 | 25.0 | 11 |
| 13 | X | 30.0 | 12 |

$259.2 \times 10^{-6}$

$1/\lambda = .00386 \times 10^{6} =$

MTBF = 3680 hrs.

### 9.2.2 Fault Code Establishment

If the information along each interdependency line is sensed, the failure of any LRU will produce a fault-code pattern. This pattern can be used to isolate the fault. In the foregoing example, if a "1" indicates no fault and a "0" indicates a fault, the code will be:

| | SENSOR NO. | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **LRU FAILED** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| No Fault | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

It can be noted from the above pattern that only those sensors that contribute to the uniqueness of the fault pattern need be used. Thus, it can be seen that sensors 2 and 3 and 7 and 8 could be removed without sacrificing the uniqueness of the words. Also, sensors 13, 14 and 15 will all fault together since LRU's 9 and 10 are connected in a loop. Thus, any two of these (such as 13 and 14) can be omitted since they supply no additional information. It is evident that a fault in LRU 9 cannot be discerned from a fault in LRU 10 by the sensor outputs only, however. The redundant sensors can be omitted only if the LRU that has multiple outputs will lose all of its outputs when it fails. If this is not certain to occur, then the sensors should be chosen on the basis of the probabilities of failure.

### 9.2.3 Interdependency Establishment

In the example of the table of failure rates shows that the outputs of LRU 2 are all equally likely to fail. A channel splitter (LRU 2) usually accepts a common input and then branches into several identical and independent channels. This means that each output could fail without affecting the others. In this case, monitoring one of the outputs would not assure the status of the others. The three synchronizer outputs, however, would not be independent. All would be derived from a common clock that would be counted down to the PRF frequency. All would have the same repetition rate so that all would depend on the complete counter working properly. If the relative complexities of the circuits needed to generate the outputs are as shown in Figure 9-5, the probability of separate or related failures of the three outputs can be calculated based on these relative complexities. The probability of detecting a LRU failure by monitoring only output number 1 expresses the dependency relationship that output 1 has with the rest of the LRU. This probability is

$$D_1 = \frac{\text{no. of common elements}}{\text{total no. of elements}}$$

or

$$\frac{48}{48 + 4} = 92.2\%$$

This means that 92.2% of the times that output 1 fails, the other outputs will be missing also. The other two outputs will have dependencies

$$D_2 = \frac{48}{48 + 6} = 89\%$$

and

$$D_3 = \frac{48}{48 + 8} = 86\%$$

In general this dependency can be calculated for any output with respect to any other output by the formula

$$D = \frac{\lambda c}{\lambda c + \lambda s} \quad \text{where}$$

D is the dependency factor

$\lambda c$ is the failure rate of the common elements between the outputs in question and

$\lambda s$ is the failure rate of those portions of the LRU peculiar to the output for which D is being calculated.

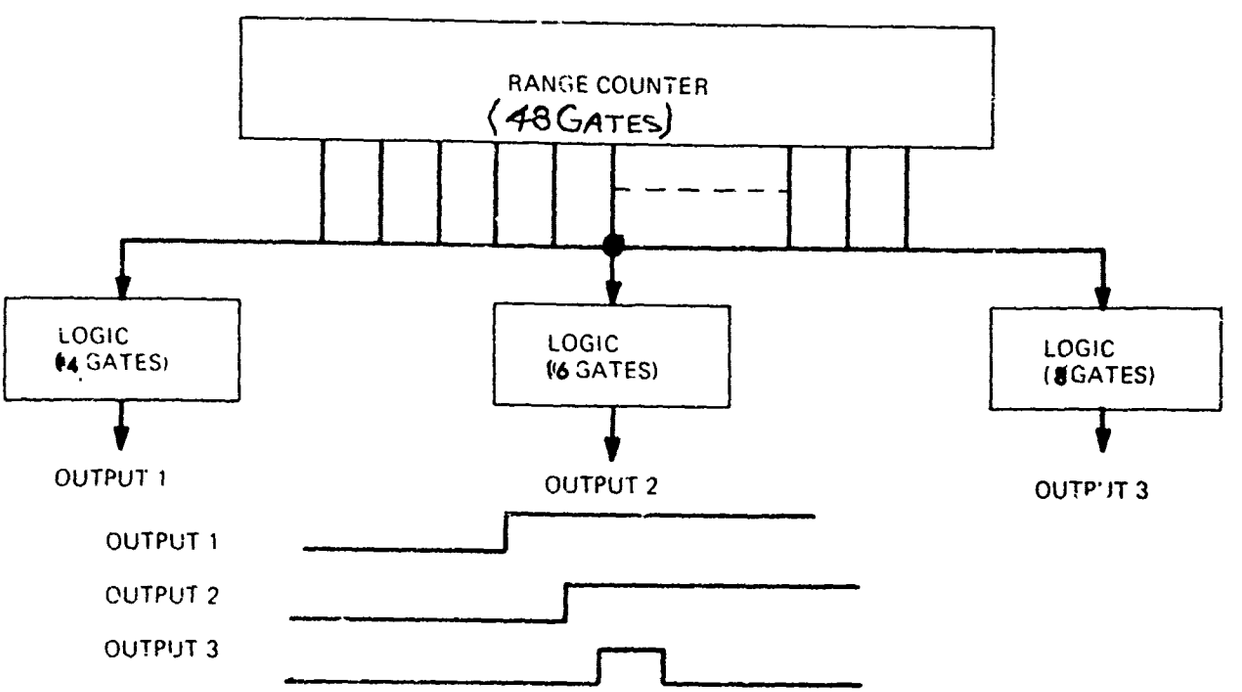



**FIGURE 9-5 EXAMPLE SYNCHRONIZER**

In the example, the failure rates ($\lambda$'s) are considered to be proportional to the number of circuits in each branch, assuming that the type of logic is identical throughout. Also in the example, the dependency of any output is the same for either of the remaining two outputs.

A system can be completely described by listing the outputs, the failure rates, and the output dependencies for each LRU. Thus, the example case is:

| Output Number | LRU Number | Output Goes to LRU Number | Failure Rate $\times 10^{-6}$ | Number of Dependencies |
|---|---|---|---|---|
| 1 | 1 | 2 | 13.1 | 0 |
| 2 | 2 | 3 | 4.3 | 0 |
| 3 | 2 | 4 | 4.3 | 0 |
| 4 | 2 | 7 | 4.3 | 0 |
| 5 | 3 | 5 | 10.1 | 0 |
| 6 | 4 | 6 | 12.6 | 0 |
| 7 | 5 | 4 | 16.7 | 2 |
| 8 | 5 | 6 | 18.0 | 2 |
| 9 | 5 | 7 | 21.3 | 2 |
| 10 | 6 | 7 | 15.3 | 0 |
| 11 | 7 | 8 | 12.0 | 0 |
| 12 | 8 | 9 | 25.2 | 0 |
| 13 | 9 | 10 | 11.3 | 1 |
| 14 | 9 | 11 | 11.3 | 1 |
| 15 | 10 | 9 | 15.6 | 0 |
| 16 | 11 | 12 | 20.1 | 0 |
| 17 | 12 | 13 | 25.0 | 0 |
| 18 | 13 | 14 | 30.0 | 0 |

And the dependencies are:

| Output Number | Percent | On Output Number |
|---|---|---|
| 7 | 92.2 | 8 |
| 7 | 92.2 | 9 |
| 8 | 89 | 7 |
| 8 | 89 | 9 |
| 9 | 86 | 7 |
| 9 | 86 | 8 |
| 13 | 100 | 14 |
| 14 | 100 | 13 |

The dependency connotations are given for each output that has a dependent association with another output. Three types of multiple output situations are illustrated. LRU 2 has three outputs with a zero dependency, i.e. each output is completely independent of the others. LRU 5 has three dependent outputs. For instance, output 7 has a 92.2% dependency with output 8. This means that 92.2% of the time that output 7 is missing because of a failure of LRU 5, output 8 will be missing also. LRU 9 has two outputs which are completely dependent, i.e., they are taken from the same point internal to the LRU. This is a 100% dependency, for whenever one output is missing, the other is certain to be missing also.

### 9.2.4 Evaluation of a Particular Sensor

With the above information, the usefulness of placing a sensor at any particular point in the system can be calculated on the basis of the uniqueness of the fault pattern produced by a given set of sensors and by the probability that a given fault will occur. In general, the more faults that a given sensor system can isolate (particularly on the first attempt), the better the system is. In selecting the optimum placement of sensors a quantitative measurement of the system capability to isolate faults is required. If the fault code pattern produced by a sensor set for a given fault is unique, then that fault will be isolated on the first attempt. If a particular fault does not produce a unique fault code pattern, the fault cannot be isolated with certainty on the first attempt. However, if the fault is isolated to two possible LRU's, the repair can be made on the second substitution. This is not as desirable as immediate isolation, but is assuredly more desirable than isolating the fault on three or greater attempts. Another factor in assessing the value of a given set of sensors would be whether or not a set isolates the most frequent faults. Thus, those sensors isolating frequent faults are more desirable than those isolating faults that occur less often. Given the time that the equipment is to operate and the failure rate of the LRU's the probability of a fault in a particular LRU during the mission time can be found by

$$P = 1 - e^{-\lambda T} \text{ where}$$

P is the probability of failure,

$\lambda$ is the failure rate per $10^3$ hours of the LRU and

T is the mission time.

### 3.5 Evaluation of a Sensor Set

A sensor set evaluation coefficient can be calculated by:

$$SE = \frac{P_1}{P} + \frac{1}{2}\frac{P_2}{P} + \cdots \frac{1}{n}\frac{P_n}{P} \text{ where}$$

SE is the sensor evaluation factor,

$P_1$ is the probability of the failures isolated on the first substitution, i.e. those failures uniquely specified by the fault code,

$P_2$ is the probability of the failures isolated after two substitutions, i.e. those failures isolated to pairs by the fault code,

$P_n$ is the probability of the failures isolated after n substitutions, i.e. those failures isolated to groups of n by the fault code, and

P is the probability of any failure.

The 1/n coefficients give less and less weight to later substitutions although the effect of adding an additional substitution decreases as n gets larger. (The weighting changes greatly when going from 1 to 2 substitutions, but less so when going from 10 to 11.)

The question of dependent outputs can be handled by considering the following simple case. The LRU depicted in Figure 9-6 is a simple resistive network having 3 dependent outputs each seeing an infinite load

impedance. Assuming all the resistors are identical and have equal dissipations, the failure rates will be the same. The output dependency relationships can be calculated from the relative complexities of the parts. Whenever output A fails there is a 50/50 chance that R1 has failed. This means that 50 percent of the time outputs B and C will be missing also. Therefore the dependency with A can be expressed by:

| Output | Dependency With | Output |
|---|---|---|
| A | 50% | B |
| A | 50% | C |

B is identically the same type of output as A so therefore:

| Output | Dependency With | Output |
|---|---|---|
| B | 50% | A |
| B | 50% | C |

When output C fails there is a 1/5 chance that the failure will be due to R1, causing A and B to be missing also. Therefore:

| Output | Dependency With | Output |
|---|---|---|
| C | 20% | A |
| C | 20% | B |

If all three outputs are sensed, (for instance, if only A is sensed) the percentage of isolated failures to the total possible failures can be calculated in the form of the sensor evaluation coefficient as:

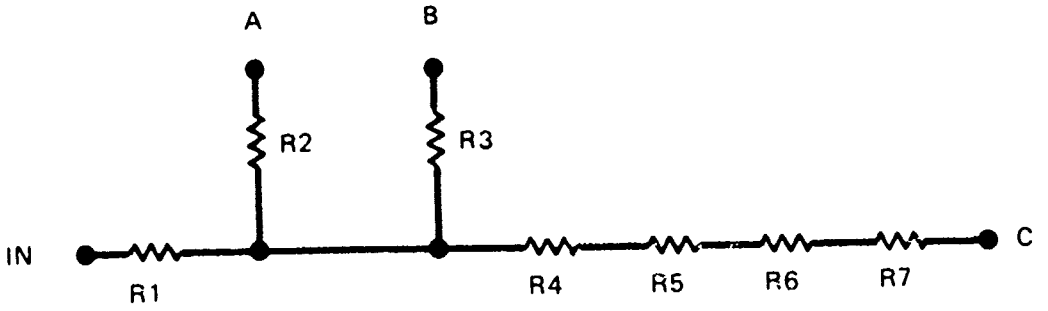



**FIGURE 9-6 ILLUSTRATIVE EXAMPLE**

This can be seen to be correct, for sensors at A and C will isolate 6/7 of the total possible failures. In general, then, the sensor evaluation coefficient can be found from the quotient of the summation of the failure probabilities of all unique fault code failures by the summation of all failures, in each case subtracting out a single dependency product wherever two dependent inputs appear (either numerator or denominator).

The total merit of any set of sensors can be evaluated by calculating a sensor evaluation coefficient for a set of sensors and comparing it with the coefficient for other sets. The higher the coefficient, the better the sensor set. Also, the relative effectiveness of any particular BIT system can be ascertained by comparing its sensor evaluation coefficient to the sensor evaluation coefficients of other systems.

## 9.3 Bit Detectability-Level Determination

This section provides the formulas and methodology to determine if the specified BIT level of a system can be met. A BIT requirement can be stated as "the BIT shall detect failure modes which represent at least "X" percent of the system failure probability".

The formula expressing this criteria is:

$$\frac{\lambda - \lambda\mu}{\lambda} \quad (100) = \text{"X"}$$

where

"X" = BIT detection probability (percentage)
$\lambda$ = the failure rate of the complete system (failures/$10^6$ hours) and
$\lambda\mu$ = the failure rate of the BIT undetected portion of the system (failures/$10^6$ hours).

The following analysis is accomplished by examining each assembly circuit schematic:

a. Determine the undetected parts. A part is considered undetected by BIT if at least one failure mode (considering opens. shorts, and out-of-tolerance conditions) cannot be detected by BIT as a failure;

b. Determine the gross failure rate of each undetected part;

c. Apportion the gross part-failure rate to each failure mode identified as undetected;

d. Choose high failure-rate items as priority interrogation candidates for the BIT system;

$$SE_A = \frac{P_A}{P} \qquad \frac{P_{R1} + P_{R2}}{P_{R1} + P_{R2} + P_{R3} \cdots + P_{R7}}$$

Assuming that the failure rates per resistor are all equal to one, this reduces to

$$SE_A = 2/7$$

Similarly,

$$SE_B = 2/7$$

$$SE_C = 4/7$$

The outputs, failure probabilities, and sensor evaluation coefficients (SEC) can be summarized as follows:

| Output | Failure Prob. (P) | Dependency (D) | On Output | SEC |
|---|---|---|---|---|
| A | 2 | 50% | B | 2/7 |
| A | 2 | 50% | C | 2/7 |
| B | 2 | 50% | A | 2/7 |
| B | 2 | 50% | C | 2/7 |
| C | 5 | 20% | A | 4/7 |
| C | 5 | 20% | B | 4/7 |

A generalized formula for the sensor evaluation coefficient for the first substitution only is as follows:

For a sensor at A only

$$SE_A = \frac{P_A}{P_A + (P_B - D_B P_B) + (P_C - D_C P_C)} = \frac{2}{2 + (2-1) + (5-1)} = \frac{2}{7}$$

For sensors at A and C we have

$$SE = \frac{P_A + P_C - P_C D_C}{P_A + P_B + P_C - P_B D_B - P_C D_C} = \frac{2 + 5 - 1}{2 + 2 + 5 - 1 - 1} = \frac{6}{7}$$

e. Sum the failure rates of the undetected part modes to obtain $\lambda_\mu$;

f. Obtain the latest failure-rate prediction for the system. This is inversely proportional to the predicted MTBF and is equal to $\lambda$.

g. Calculate

$$\frac{\lambda - \lambda_\mu}{\lambda}$$

An example of the type of worksheet necessary to compile the undetected failure modes is shown in Table 9-2. Failure mode determination is described in Section 6 of this Design Guide.

For complex systems, computer programs exist which can be adapted to display component-failure rates associated with faults and undetected modes.

**9.4 Relationship Of Bit Level To Availability:** In attempting to pursue a BIT-level optimization approach, we are faced with two undefined parameters: BIT level, which is the dependent variable; and, availability, with definite one of these two parameters, we would have an infinite number of solutions to BIT level. Availability is the most influential parameter which controls the reliability-availability-detectability relationship. Allowing availability to remain undefined causes ambiguities and inconsistencies.

Availability, as used in the equations in this Guide, is defined in MIL-STD 721B as: a measure of the degree to which an item is in the operable and committable state at the start of a mission, when the mission is called for at an unknown (random) point in time. The relationship of mean corrective-maintenance time, mean-time-between-failures, and availability can be shown in the nomographs in reference 111. Maintainability demonstration tests are described in reference 103 of the specifications listed in the bibliography.

The question of whether BIT detectability level should be 0.85, or 0.75 or some other level, should be governed by the system availability requirement. The mean corrective-maintenance downtime ($M_{ct}$) would be set by first establishing the BIT detectability level. In order to arrive at the detectability level, certain parameters should be considered.

Availability (A) of a weapon system is a complex function of equipment mean-time-between-failures (MTBF), mean corrective-maintenance time ($M_{ct}$), duty cycle (d), failure detectability (k) of checkout and test equipment, mission time ($t_m$), and the functional performance threshold ($G_{min}$) at which the system can be classified as operationally ready for a particular mission assignment. Availability can be represented symbolically as a function of these parameters as follows:

$$A = f(MTBF, M_{ct}, d, k, t_m, G_{min}).$$

| Subassembly | Ckt Designation | Undetected Mode | $\lambda\mu / 10^6$ Hrs |
|---|---|---|---|
| Chassis | DS1 | O | 2.500 |
| | DS2 | O | 2.500 |
| Power Supply | R2 | O, S, D * | 0.001 |
| | R4 | S, D | 0.002 |
| | R9 | O | 0.132 |
| | R10 | S | 0.001 |
| | R11 | O | 0.002 |
| | R16 | S | 0.001 |
| | R17 | S | 0.001 |
| | R18 | O | 0.008 |
| | R20 | O | 0.008 |
| | R21 | O | 0.008 |
| | R22 | S | 0.001 |
| | R23 | O | 0.008 |
| | R24 | O | 0.008 |
| | R25 | S | 0.001 |
| | C16 | S | 0.001 |
| | C1 | S | 0.210 |
| | C2 | S | 0.210 |
| | C3 | S | 0.210 |
| | C9 | S | 0.210 |
| | C11 | S | 0.204 |
| | C12 | S | 0.204 |
| | C13 | S | 0.001 |
| | C14 | S | 0.001 |
| | C15 | S | 0.001 |
| | CR16 | S | 0.019 |
| | Q1 | S, O | 0.307 |
| | Q2 | O | 0.082 |
| | U1 | S, O | 2.760 |
| | U2 | S, O | 2.760 |
| | ⋮ | ⋮ | ⋮ |

$\Sigma \lambda \mu =$

* S = Short, O = Open,
D = Degraded (Out-of-Tolerance)

**TABLE 9-2 BIT UNDETECTED PARTS EXAMPLE**

The discussion which follows on the relationship of detectability, reliability, and availability is edited from Navord OD 39223, Maintainability Engineering Handbook.

"Figure 9-7 illustrates the relationship between maintainability and the established availability and reliability requirements in a system for which a maximum allowable turnaround time for maintenance is specified. Availability (operational readiness) of the system at time ($t_a$) is given by:

$$A(t_a) = R(t_m) + (kM(t_r))(1 - R(t_m)),$$

where,

$R(t_m)$ = Probability that the system will survive the specified mission of duration $t_m$ without failure,

$t_r$ = Specified turnaround time, or maximum downtime for repair required of the system,

k = Probability that if a system failure occurs it will be detected during the mission, or during system checkout following the mission, and

$M(t_r)$ = Probability that a detected system failure can be repaired in time, $t_r$, to restore the system to operational status.

When mission reliability, mission duration, availability, and turnaround time are specified for the system, the detectability-times-maintainability function for the system is constrained to pass through or exceed the point given by:

$$KM(T_r) \geq P(At_a),$$

where

$$M(t_r) > P \text{ and}$$

$$P \geq P_r(\text{System is up at } t_a / \text{System is down at } t_a - t_r)$$

Consider the example of the following specified operational characteristics for a new ASW system:

Mission reliability ($R(t_m)$) = 0.80 for $t_m$ of 8 hours; and

Mission availability ($A(t_r)$) = 0.95 for turnaround time, $t_r$, of 30 minutes following completion of a preceding mission or initial receipt of an alert.

From these requirements, the required detectability-maintainability product (kM) is derived as follows:

$$kM(.5) = \frac{A(9.5)-R(8)}{1-R(8)} = \frac{0.95-0.80}{1-0.80} = 0.75.$$

Therefore, kM (.5) = 0.75 is the joint probability, given that a system failure has occurred, that the failure will be detected (either during the mission or during post-mission checkout) and will be repaired within 30 minutes following completion of the mission. Some trade-off potential exists between k and M, as illustrated in Figure 9-8.

Assume, for example, that the k requirement is to be set at 0.9, i.e., built-in system readiness,

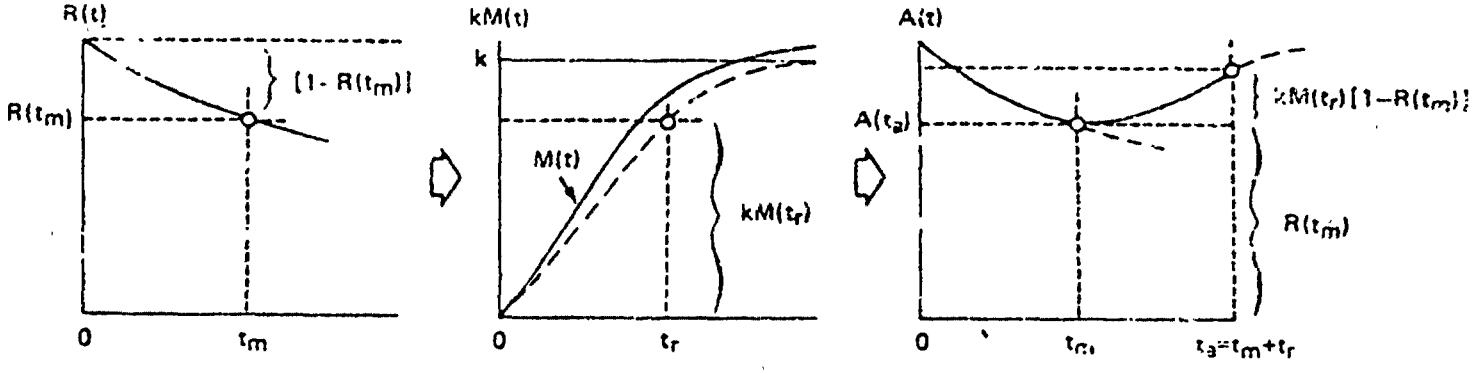



**FIGURE 9-7 AVAILABILITY AS A FUNCTION OF TIME.**

measurement features are to be incorporated in the design to detect at least 90 percent of the system failures and to provide the system operator with GO/NO-GO indications of system readiness. In translating this requirement to a specified design requirement, the following criteria should be called out:

(a) Failure Detectability (k). Specify the required probability of system failure detection (e.g., k = 0.90) or proportion of predicted system failures (approximately weighted according to relative failure rate) which must be detected and translated to system NO-GO indication;

(b) Performance Level — Define the minimum level of system performance with which the mission can be accomplished and below which the readiness indicator must provide a NO-GO indication; and

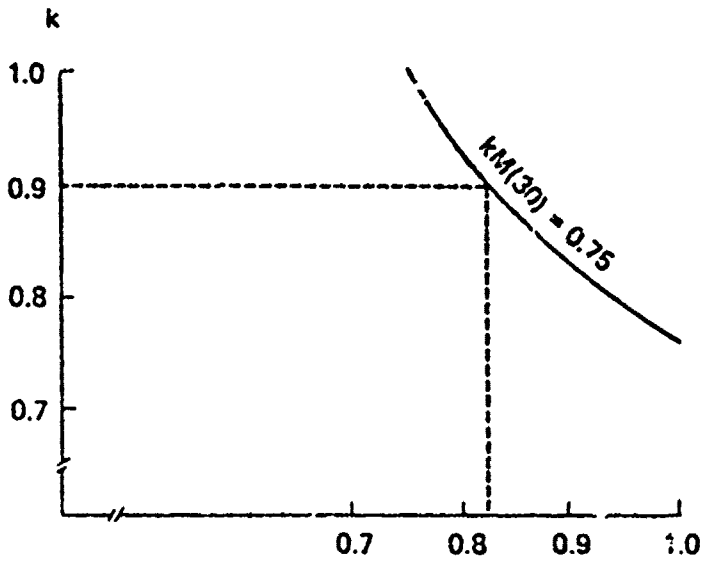



**FIGURE 9-8 TRADE-OFF POTENTIAL BETWEEN FAILURE DETECTABILITY AND MAINTAINABILITY**

(c) Mode of Operation — Specify whether built-in-test features are to provide automatic continuous monitoring and GO/NO-GO indication, or are to be manually operated to provide the performance check and GO/NO-GO readout on demand.

The maintainability requirement for this example is then derived as follows:

$$M(30) = \frac{0.75}{0.90} \approx 0.83.$$

This means that 83 percent of all system repair actions detected during the mission or during post-mission checkout must be completed within the specified 30-minute reaction time.

"Using the exponential approximation, maintainability as a function of repair time is expressed as the probability of repair in time $t_r$:

$$M(t_r) = 1 - e^{-\mu t_r} = 1 - e^{-t_r/M_{ct}}$$

where

$\mu$ = repair rate, $1/M_{ct}$, and

$t_r$ = repair time for which M(t) is to be estimated.

This required mean time to repair ($M_{ct}$) is found from the following equation:

$$M_{ct} = \frac{t_r}{\ln(1 - M(t_r))}$$

Substituting $t_r$ = 30 minutes, $M(t_r) = 0.83$, gives:

$$M_{ct} = \frac{30}{-\ln(0.17)} = \frac{30}{-(-1.77)} = 17 \text{ min.}$$

And, for $M(t_{max}) = 0.95$, we find the maximum time for repair of 95 percent of detected system failures ($M_{max_{ct}}$) as follows:

$$M(t_{max}) = 0.95 = 1 - e^{-M_{max_{ct}}/M_{ct}}, \text{ and}$$

$$M_{max_{ct}} = -M_{ct}(\ln(1-0.95)) = -(17)(-3) = 51 \text{ minutes.}$$

## 9.5 An Approach to Bit Optimization

The following guidelines are provided in order to arrive at an optimum BIT level for an initially designed system. The many variables that must be considered when deciding upon the optimum BIT level include cost, size, speed of operation, reliability, and corrective-maintenance time. It is advantageous for BIT to be considered from the initial design phase. If BIT considerations are left to the final design-cleanup phase, they are likely to be more costly and less effective.

### 9.5.1 Corrective Maintenance Time ($M_{ct}$ Calculations)

In the BIT optimization process, it is necessary to calculate the mean corrective-maintenance time. The general approach for calculating mean corrective maintenance time follows the procedure outlined in MIL-HDBK 472, "Maintainability Prediction." Procedure II, Part B, calls for the calculation of $M_{ct}$ by applying the formula:

$$M_{ct} = \frac{\Sigma(\lambda i \, M_{cti})}{\Sigma \lambda i}$$

where

$\lambda i$, = part failure rate (failures per $10^6$ hours) and

$M_{cti}$ = corrective maintenance time.

Corrective maintenance time is the sum of the following applicable maintenance tasks (some of which can be reduced by BIT are starred):

*(a) *Localization* or determining the presence of a failure to the extent possible;

*(b) *Isolation* or determining the location of a failure to the extent possible;

(c) *Disassembly* or equipment disassembly to the extent necessary, to gain access to the item that is to be replaced;

(d) *Interchange* or removing the defective item and installing the replacement;

(e) *Reassembly* or closing and reassembly of the equipment after the replacement has been made;

*(f) *Alignment* or performing any alignment, minimum tests and/or adjustment made necessary by the repair action (if required); and

*(g) *Check out* or performing the minimum checks or tests required to verify that the equipment has been restored to satisfactory performance.

For the purpose of BIT optimization, in comparing a given subsystem with varying BIT requirements, parts and cards introduced into the subsystem to meet BIT requirements affect the calculation of the total failure rates, repair times, and related factors. The failure rate ($\lambda$) of these parts and cards is included in the total subsystem failure rate and MTBF.

Two main distinctions are made in the maintenance task times between those failure modes which are detected and which are not detected by BIT:

(a) *Fault-Localization Time*. If detected by BIT, time to localize a failure is instantaneous and is considered "0". Otherwise, if not detected by BIT, the localization portion of the maintenance task time is the same as for a system with no BIT requirements; and

(b) *Fault-Isolation Time*. The number of cards in the fault-isolation set is the determining factor in the calculation of fault-isolation time. The total isolation time is a function of the number of cards in the fault-isolation set divided by the total number of cards in the assembly under consideration. Thus, when fault-isolation does not go below the assembly level, this ratio equals one, and there is no improvement in fault isolation time.

To summarize,

| | BIT Detected | No BIT |
|---|---|---|
| Location Time | 0 | $T_L$ |
| Isolation Time | $\frac{FI}{N} T_I$ | $T_I$ |

(Maintenance-task times for disassembly, interchange, reassembly and checkout do not change regardless of what BIT level or fault-isolation card level is adopted.)

Taking into consideration BIT level and fault isolation, the formula for $M_{ct}$ becomes:

$$M_{ct} = \frac{\Sigma \lambda T_{ri} + \Sigma \lambda j T_{rj}}{\Sigma \lambda i + \Sigma \lambda j}$$

where

$\lambda i$ = failure rate for card detected by BIT,
$Tri$ = time to repair card detected by BIT,

$\lambda j$ = failure rate of card not detected by BIT, and
Trj = time to repair card not detected by BIT.

This formula is used in the BIT optimization process described in Section 3.7.3.3.

### 9.5.2 Cost Calculations

The cost data necessary to conduct the BIT optimization study are collected from production experience and include BIT hardware-fabrication costs, costs associated with connectors and test points, required auxiliary equipment and maintenance-manual preparation. A sample cost work sheet is shown in Table 9-3.

### 9.5.3 Optimization Calculations

To optimize BIT level consistent with parameters within the system, a suggested approach is to determine a BIT detection level and fault-isolation card set which results in the minimum cost that meets the mean

BIT DETECTION LEVEL: _____ Percent

FAULT ISOLATION SET: _____ Cards, modules, etc.

| COST ELEMENTS | NON-RECURRING COST | RECURRING COST |
|---|---|---|
| SHIPPING COSTS | $_____ | $_____ |
| FAULT DETECTION | _____ | _____ |
| TEST EQUIP MAIN COSTS | _____ | _____ |
| CONNECTORS TEST PTS. | _____ | _____ |
| PERSONNEL | _____ | _____ |
| AUXILIARY TEST EQUIP | _____ | _____ |
| SPARES | _____ | _____ |
| MAINTENANCE MANUAL | _____ | _____ |
| TRAINING | _____ | _____ |
| INVENTORY COSTS | _____ | _____ |
| TOTAL | $_____ | $_____ |

**TABLE 9-3 BIT TRADE-OFF COST WORKSHEET**

corrective-maintenance time ($M_{ct}$) requirement. There are two parts to the optimization process:

a. Determine various combinations of BIT levels and fault-isolation card sets which meet the $M_{ct}$ requirement; and

b. Determine BIT level corresponding to various costs for different card sets and choose one of these combinations which results in a minimum cost.

Expanding the steps to be used in the trade-off study:

a. Compute $M_{ct}$ of the system, assuming no BIT requirements and fault-isolation only to the LRU level using the procedure described in Section 9.5.1. Provide a column for each applicable maintenance task, including a separate column for "Localization and Isolation". Table 9-4 illustrates this calculation;

b. Compute $M_{ct}$ for different combinations of BIT detection level/fault-isolation card set using the procedure described in Section 9.5.1. Table 9-5 illustrates this calculation;

c. Prepare a graph which relates $M_{ct}$ and BIT level and fault-isolation card set. This graph, a sample of which is shown in Figure 9-4 depicts all combinations of BIT-level card sets which meet the $M_{ct}$ requirement. These combinations become candidates for cost analyses;

d. Prepare a cost work sheet as described in Section 9.5.2 for each candidate combination;

e. Using the cost totals arrived at, prepare a plot, similar to the example in Figure 9-9, relating costs, BIT-level/fault-isolation card set and $M_{ct}$. $M_{ct}$ is the dotted line of constraint, above which any combination of BIT level and fault-isolation card set which produces minimum cost is an acceptable recommendation.

## 9.6 Quantification Of Testability

In the design of electronic circuits for mass production, it is often important to be able to predict whether or not the probable faults can be diagnosed at the available test points. The same reasoning holds true of the design of BIT into a system. It is thus useful to define a measure of "testability", to be denoted by $M_{ct}$ which satisfies the following criteria:

(a) $M_t$ gives a unique measure of how readily element faults can be diagnosed from test terminal measurements $0 \geq M_t \geq 1$ and

(b) $M_t$ can be economically computed from the nominal circuit and its test points.

Such a definition of $M_t$, along with the algorithm needed to compute it, can be derived. The algorithm also gives a clear indication of how to add additional test points to a circuit which has a low testability.

Consider the circuit shown in Figure 9-10. Assume that N is a nonlinear circuit which is in a steady state under DC excitation. The extension to AC analysis is straightforward. Let $\hat{N}$ be the adjoint network of N (Reference 42 in the bibliography). As described in the reference, $\hat{N}$ has the following properties:

(a) $\hat{N}$ has the same topology as N;

(b) $\hat{N}$ is a linear circuit in which the nonlinear resistive elements of N have been replaced by linear resistors calculated from the incremental values of resistance of the corresponding elements of N (Figure 9-11);

| | ($f/10^6$ hrs) | LOCATION AND ISOLATION ($T_{LI}'$) | DISASSEMBLY INTERCHANGE AND REASSEMBLY | CHECKOUT | REPAIR TIME ($T_R$) | $\lambda T_R$ |
|---|---|---|---|---|---|---|
| WCIP | 389 | 5.50 | 0.50 | 0.25 | 6.25 | 2431 |
| DP | 479 | 0.50 | 0.20 | 0.10 | 0.80 | 383 |
| DCU | 556 | 4.75 | 0.25 | 0.25 | 5.25 | 2920 |
| UBSWBD | 87 | 1.25 | 0.25 | 0.08 | 1.58 | 137 |
| 2A11 | 50 | 1.00 | 1.00 | 0.25 | 2.25 | 113 |
| DSC | 107 | 1.00 | 3.00 | 0.08 | 4.08 | 437 |
| REMAINDER | 87 | 0.75 | 0.75 | 0.16 | 1.66 | 144 |

$\Sigma\lambda = 1755$

$\Sigma\lambda T_R = 6564$

$M_{ct} = 3.74$ hours

**TABLE 9-4 SCLS REPAIR TIMES**

| $T_{LI}'$ | | FI/N | | 50 Percent | | 75 Percent | | 90 Percent | | 95 Percent | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WCIP | DCU | WCIP | DCU | WCIP | DCU | WCIP | DCU | WCIP | DCU | WCIP | DCU |
| 5.5 | 4.75 | 77/77 | 105/105 | | | | | | | | |
| 5.5 | 4.75 | 32/77 | 32/105 | 2.52 | 1.91 | 3.06 | 1.38 | 2.11 | 1.35 | 2.19 | 1.39 |
| 5.5 | 4.75 | 16/77 | 16/105 | 1.95 | 1.55 | 1.20 | 0.84 | 1.08 | 0.70 | 1.10 | 0.70 |
| 5.5 | 4.75 | 8/77 | 8/105 | 1.66 | 1.37 | 0.77 | 0.57 | 0.57 | 0.37 | 0.56 | 0.36 |
| 5.5 | 4.75 | 4/77 | 4/105 | 1.32 | 1.28 | 0.56 | 0.43 | 0.31 | 0.21 | 0.29 | 0.18 |

**TABLE 9-5 TIME REQUIRED TO LOCATE AND ISOLATE A FAILURE AIDED BY BIT ($T_{LI}$)**

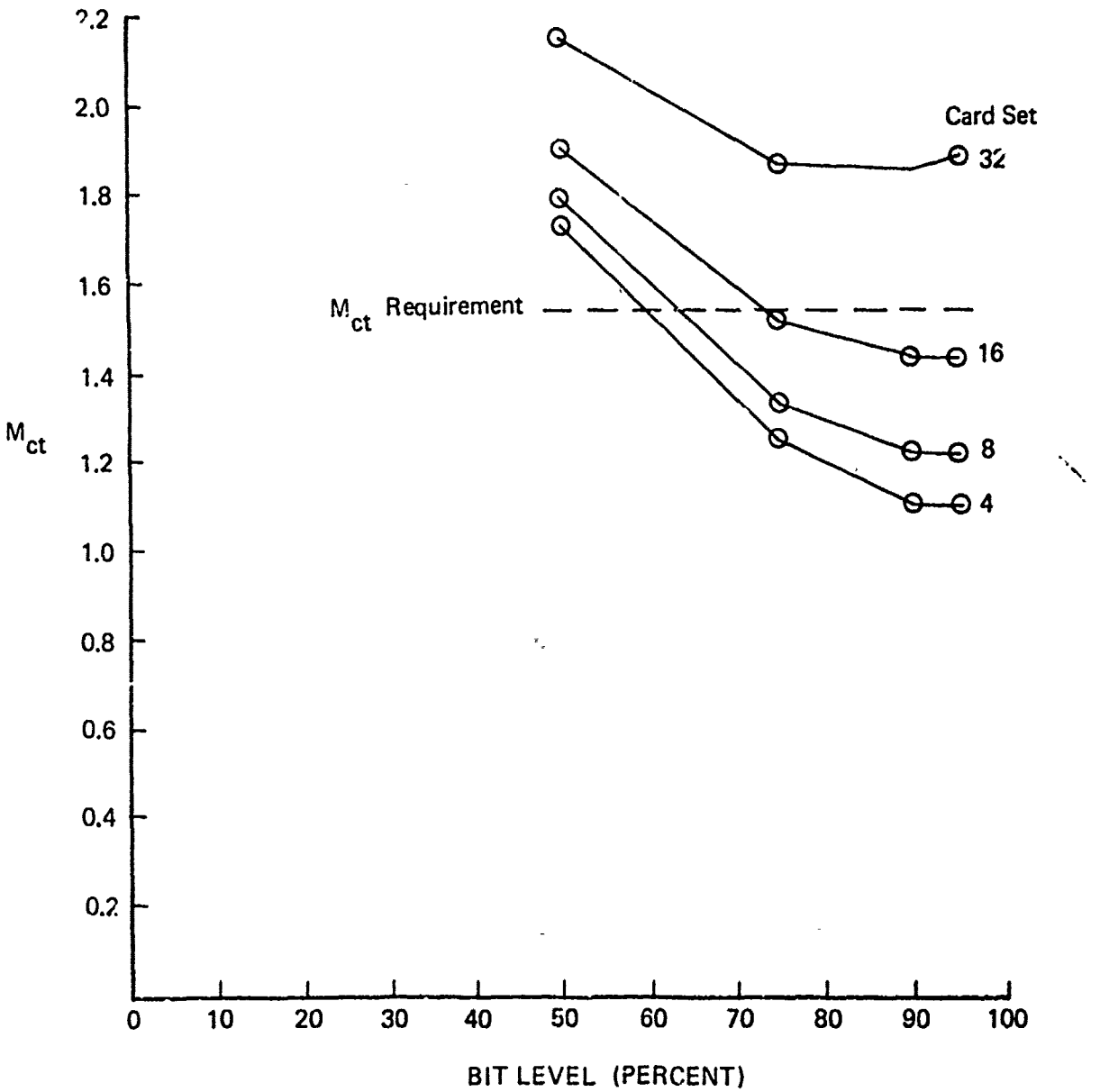



**FIGURE 9-9 BIT LEVEL TRADE-OFF EXAMPLE.**

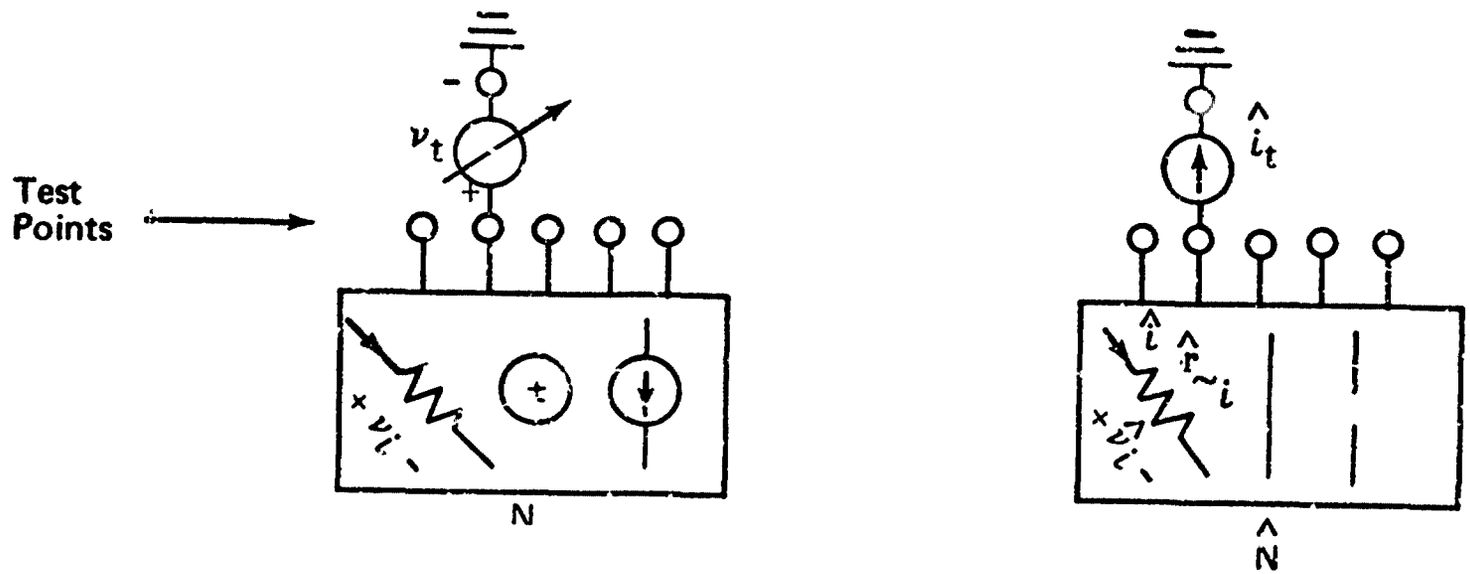



**FIGURE 9-10 TESTABILITY CIRCUIT**

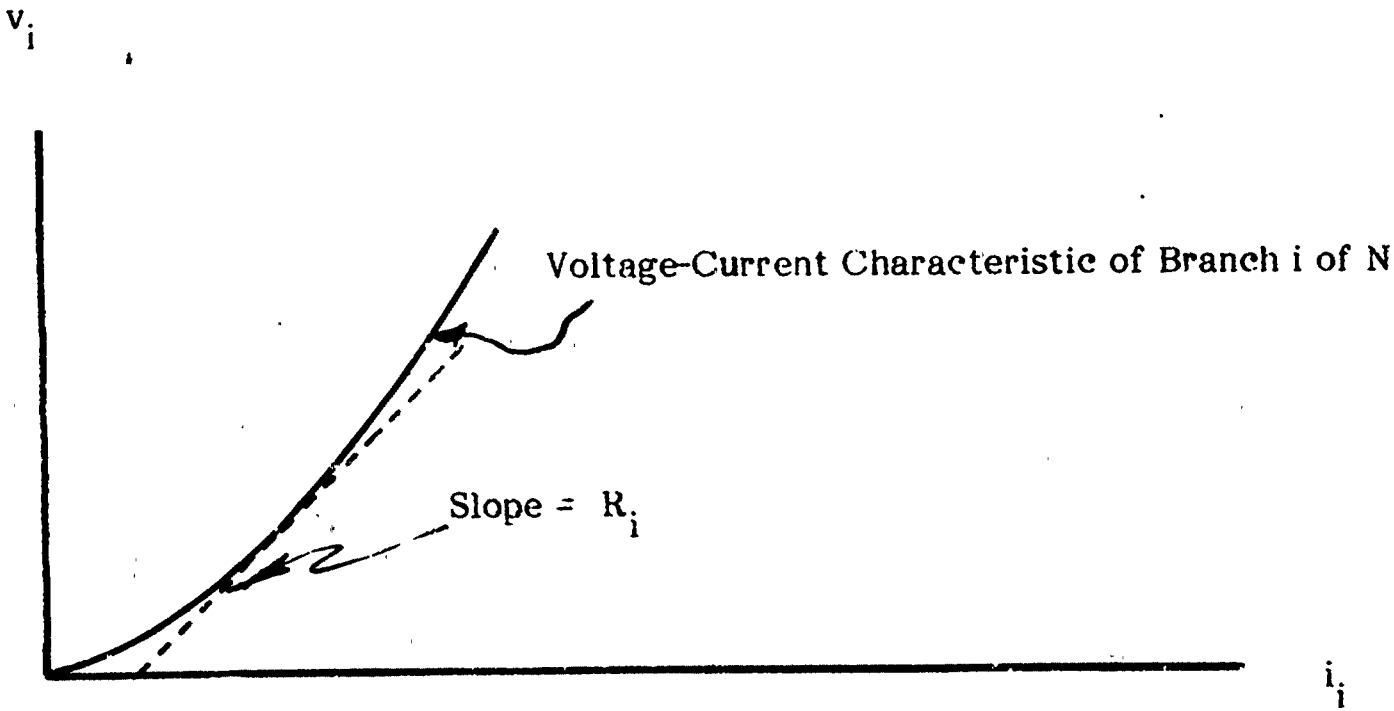



**FIGURE 9-11 VOLTAGE-CURRENT CHARACTERISTIC**

(c) All voltage and current sources of N are shorted and opened, respectively, in $\hat{N}$ (Figure 9-10); and

(d) All active elements of N are to be replaced by linear active transports in $\hat{N}$. Specifically, let a nonlinear controlled source in N be given by the voltage current relations,

$$i_1 = i_1(V_1, V_2), \text{ and}$$

$$i_2 = i_2(V_1, V_2).$$

The corresponding linear twoports in $\hat{N}$ will then be described by the equations;

$$\hat{i}_i = \frac{(\partial i_1)}{(\partial v_1)}\hat{v}_1 + \frac{(\partial i_2)}{(\partial v_1)}\hat{v}_2, \text{ and}$$

$$\hat{i}_2 = \frac{(\partial i_1)}{(\partial v_2)}\hat{v}_1 + \frac{(\partial i_2)}{(\partial v_2)}\hat{v}_2$$

which can be realized by two linear-controlled sources and resistors as indicated in Figure 9-12.

Then, it can be shown that if the second test terminal (Figure 9-10) has the nominal voltage $V_t$ and the corresponding terminal in $\hat{N}$ is excited by a current $i_t = 1A$, the relation between the small parameter changes $\Delta R_j$, $\Delta R_{Lm}$, $\Delta G_{jb}$, $\Delta\alpha_{np}$, and $\Delta\mu_{qr}$ and the corresponding change in $V_t$ is:

$$\Delta V_t = -\sum_i \hat{i}_i i_i \Delta R_i - \sum_{Lm} \hat{i}_i \Delta R_{Lm} + \sum_{j,k} \hat{v}_j v_k \Sigma G_{jk} + \sum_{n,p} \hat{v}_n i_p \Delta\alpha_{np} - \sum_{q,r} \hat{i}_q v_r \Delta\mu_{qr}$$

Here, $\Delta R_{Lm}$ represents the change in the gain of a current-controlled voltage source in branch $m$, controlled by the current of branch $L$. $\Delta G_{jk}$, $\Delta\alpha_{np}$ and $\Delta\mu_{qr}$ have similar interpretations. For nonlinear elements, these expressions are readily generalized to give $\Delta V_t$ in terms of the changes of the parameters (saturation currents, temperature, etc) of these elements. We conclude that for small changes, $\Delta_{Pi}$ in all circuit parameters $p_i$, the relation:

$$\Delta V_t \cong \sum_i S_i \Delta p_i$$

holds. The sensitivity $S_i$ can be computed from the branch currents and voltages of the nominal network N under its actual excitations and the *linear* circuit $\hat{N}$ excited at the test terminal where $V_t$ is measured. Assume next that the sensitivities $S_{ki}$ of all test-point voltages $V_{tk}$, $k = 1,2.....,n$ are required, where n is the number of test terminals. Since the branch variables of N, needed in the calculation of $S_{ki}$, are found under nominal element values and excitations, N has to be analyzed only once. By contrast, the linear network $\hat{N}$ must be analyzed $n$ times, with a different test terminal excited by the 1A source each time. However, the computational effort can be greatly reduced compared to that needed for $n$ separate analysis if matrix inversion rather than Gaussian elimination is used in the analysis. Assume for simplicity, that $\hat{N}$ can be analyzed using nodal analysis. Since it has no internal independent sources, it can be characterized by the equations:

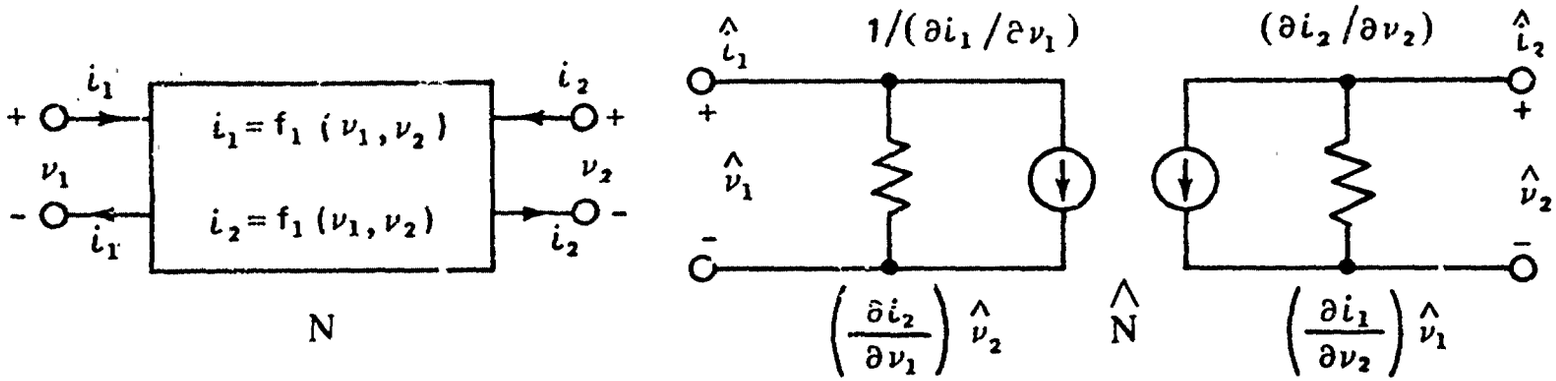



**FIGURE 9-12 TWO LINEAR-CONTROLLED SOURCES**

$$\hat{Y}_n \hat{V}_n = \hat{i}_t$$

Here $\hat{Y}_n$ is the nodal admittance matrix, and $\hat{V}$ the node voltage vector from which all branch variables of $\hat{N}$ are readily found. Finally, $\hat{i}_t$ is the vector of excitations which is now one of the $n$ possible vectors.

$$\underbrace{(1, 0, 0, \ldots, 0}_{n}, \underbrace{0, \ldots, 0)^t}_{m-n}$$

$$(0, 1, 0, \ldots, 0, 0, \ldots, 0)^t$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\underbrace{(0, 0, 0, \ldots, 1, 0, \ldots, 0)^t}_{m}$$

Here we assumed that N (and hence $\hat{N}$) has $m$ modes so ordered that the $n$ test points ($n \leq m$) are numbered 1,2,....,n, and the rest of the modes n + 1,...,m. Computing the inverse $\hat{Y}_n^{-1}$ of $\hat{Y}_n$, we have

$$\hat{V}_n = \hat{Y}_n^{-1} \hat{i}_t.$$

Hence, as can be seen from the special form of $i_t$, $\hat{V}_r$ is simply one of the first $n$ columns of $\hat{Y}_p$. The location of the nonzero element in $\hat{i}_t$ determines which column gives $\hat{V}$. Thus, finding $\hat{V}_n$ from $\hat{Y}_n$ $-1$ needs no additional operations. The derivation was performed from the case when nodal analysis was possible for $\hat{N}$. It can be readily extended for the more general case when hybrid analysis is used. We conclude that all branch variables of N under *all* excitations at their test terminals can be computed at little more cost than that required to invert the m x m matrix $Y_n$. Since the calculation of the $S_{ki}$ needs only the branch variables of N and $\hat{N}$, *all* sensitivities of *all* test voltages with respect to *all* parameters in N can be obtained at the cost of one analysis of N with nominal parameter values and excitations, *plus*, the inversion of an m x m matrix, *plus* some trivial operations. The analysis of the nominal circuit N is necessary anyway. For nonlinear elements in N, the cost of the matrix inversion is negligible compared to the cost of the analysis of N.

Assume now that all sensitivities $S_{ki}$ have been found. Let the minimum voltage change which the meter can detect at a test point be $\delta$ V, and let the tolerance of the ith circuit parameter be $\Delta p_i$. We can, therefore, detect an out-of-tolerance element if

$$\left|\Delta v_k\right| \cong \left|S_{ki} \Delta p_i\right| \geq \delta V,$$

for some test point voltage $V_k$. This condition is readily checked from the given $S_{ki}$ and $\Delta p_i$ values. Let the number of parameters $p_i$ for which the inequality holds be $n_1$, while the total number of parameters = $n_t$. The equation

$$M_t \triangleq \frac{n_i}{n_t}$$

gives an easily computed measure of the testability of the circuit. The adjoint network, in addition to helping

in the calculation of the $S_{ki}$, is also helpful in showing *where to add more test points* if the circuit is not testable. Since the sensitivity of any parameter $p_i$ is proportionate to $\hat{V}_i$ or $\hat{i}_i$, the effective place for an additional terminal is at a node where it will cause a large $V_i$ or $i_i$ to appear. Since $\hat{N}$ is a linear network, it is relatively easy to locate such points for any parameter $p_i$. The new sensitivities can be found without an appreciable amount of additional calculation, since the new node-voltage vector $\hat{V}_n$ is simply a new (hitherto unused) column of $\hat{Y}_n - 1$.

The above discussions immediately suggest a constructive approach to finding the test points. This consists of finding $\hat{Y}_n - 1$ as before, and regarding each column as a node-voltage vector $\hat{V}_n$. Next, the branch variables of $\hat{N}$ are computed, and from these, the $m$ sets of sensitivities $S_{ki}$, K = 1,2,.....,m. Then, the first test point is chosen as the one for which the largest number of parameters $p_i$ meets the condition $S_{ki}\Delta p_i \geq \delta V$. The second node to be chosen as a test point is the one which causes the largest number of the remaining parameters to meet the testability condition. The process ends when all parameters are testable at some test point. So far we have not touched on the question of *uniqueness* in testing. By observing the signs, as well as the sizes, of the sensitivities, it is possible to find test points which satisfy uniqueness as well as testability requirements.

Another more important limitation of the above method concerns the validity of the sensitivity-oriented approach for large changes in the parameter values. Clearly, this gives, at best, a good approximation of the true changes in test-point voltages. The validity of the result is, therefore, restricted to relatively small tolerances. However, if a small deviation is testable then, in all likelihood, so is a large one. Hence, the described approach can be used to get a sufficient (if not necessary) condition for testability.

**9.7 Formal Approaches To BIT:** There have been a number of attempts to establish a formal basis for BIT. These have taken the form of:

a. Efficient algorithmic means for testing digital circuits;

b. Diagnostic and fault-localization procedures; and

c. Redundancy techniques and fault-tolerant designs.

The last category applies to systems which are not accessible for servicing or which demand a level of reliability wherein the cost of the additional hardware is justified by the mission requirements. These are not repairable systems in terms of BIT applications.

Theoretical studies of the first two categories are reminiscent of the earlier attention to the logic-minimization techniques by means of Boolean algebra. These techniques (the Karnaugh map or the Quine McClusky Table) offer means of reducing second-order Boolean expressions to a minimum number of terms of a minimum number of variables which can be implemented with a minimum of gates. With two levels of NAND logic, the implementation is minimal as well. This, however, covers only part of the digital circuit. The memory elements have not lent themselves to convenient analysis. The studies of state, assignment, minimization, and partitioning have enjoyed more academic than practical attention.

The two types of circuits normally used in digital circuits (computers, switching systems, and control systems) are combinational circuits and sequential circuits. Combinational circuits have their output values at a given time dependent only upon their present inputs and are characterized by the absence of feedback (closed loops). Sequential circuits have their output values at a given time dependent not only upon present inputs but also upon inputs applied previously.

Studies such as Sellers (84)* only deal with simple combinational elements and are totally inadequate for the MSI and LSI devices now in use. The theoretical work on sequential machines (combinational and memory elements) is more sparse and less relevant.

For combinational circuits (1, 4, 21, 26, 44, 50, 56, 92) an example of a procedure for generating test sets for large circuits is the utilization of the path-sensitizing algorithm (99) and the D-Algorithm (45, 67, 69).

The path-sensitizing method is used for manual testing and computer solutions to test problems. It uses the normal-form representation of the Boolean function satisfied by the circuit to be tested. "Each gate is first assigned a number representative of its logic level. The Boolean equation of the circuit is then derived and, as the derivation proceeds, all signal paths through the network are identified. The final equation indicates the input vector which will propagate a signal through any path in the circuit. From the sensitizing of all paths and the observation of the circuit outputs, the presence of any fault can be detected and the location of the fault can be determined to within a small number of gates, if not to the level of a single gate."

"The D-Algorithm method is based upon the cubical complex description of logic functions. This method is used for the computer derivation of tests for large and small networks. A cover is formed over the network whose cubes are intersected with each other to form D-cubes. The D-cubes indicate how each gate output is reflected in signals throughout the circuit. Successive intersections are made with the cubes until a set of intersections is found which causes the selected gate output to appear as a determinant variable in the output function of the circuit. The final cube formed in the series of intersections is then intersected with the network cover to find the input vector which will serve as a circuit test input for the given gate. This method of generating sequential circuit tests is difficult to apply and yields very large sets of tests. However, it is used in tests where it is not possible to employ hardware BIT."

*NOTE: Reference numbers refer to NELC Bibliography in Section 10.

Test sequences for sequential circuits (32, 52, 53, 54, 57, 63, 67, 69, 78) are represented by state tables (flow tables). The flow table is a tabular means by which the requirements of a sequential circuit may be stated precisely and by which redundancy in these requirements may be recognized and eliminated. Cod in binary form, are assigned to describe the internal states of the sequential circuit (50) so that the circ... output state is described for all possible sequences of input states. The models are described by Mealy (1955) and Moore (1956)(69) and bear their names.

BIT must relate to current design practices and, since there are no criteria for determining an optimum design, the approach to BIT is strongly investigative. Experience and common sense are important design parameters. There may be, therefore, many valid approaches to BIT. Some uniformity can be achieved by the prescribed use of fault tables and by the use of standard microprocessor functional modules.

# SECTION 10

# BIT REFERENCES

**10.1** The primary documents used in preparing this interim design guide were:

a. NAVMATINST 3760.9, *Built-In-Test (BIT) Design Guide,* Naval Electronics Laboratory Center (now Naval Ocean Systems Center), San Diego CA, 92152.

b. *A Study of a Standard BIT Circuit,* Feb 77, Final Report by Research Triangle Institute for the Naval Avionics Facility, Indianapolis IN, 46218.

c. RADC-TR-71-281, *Design of Integral Sensor Test System,* Dec 71, Rome Air Development Center, Griffiss AFB NY, 13441, (AD-890479L).

The NELC Design Guide provided an extensive bibliography. This is presented below for the convenience of the reader.

## 10.2 NELC References and Bibliography

### a. BOOKS


1. Chang, H. Y., Manning, E., Metze, G., *Fault diagnosis of digital systems,* Wiley-Interscience, (1970)

2. Coombs, C. F., *Basic electronic instrument handbook* (1972)

3. Erlinger, M. A., *Microprogramming and its use as an extensible processor* (1972)

4. Friedman, A. D., Menon, P. R., *Fault detection in digital circuits,* Prentice-Hall, (1971)

5. Giles, A. F., *Electronic sensing devices* (1968)

6. Goldman, A., *Maintainability, a major element of systems effectiveness* (1969)

7. Jensen, R. W., Lieberman, M. D., *IBM electronic circuit analysis program* (1970)

8. Karpus, W. J., Soroka, W. W., *Analog methods* (1959)

9. Kayton, M., Fried, W. R., *Avionics of navigation systems* (1969)

10. Landee, Davis, Albrecht, *Electronic designers' handbook* (1970)

11. McCray, J. A., Cahill, T. A., *Electronic circuit analysis for scientists* (1973)

12. Meisel, W. S., *Computer oriented approaches to pattern recognition* (1972)

13. Morrow, L. C., *Maintainability engineering handbook* (1957)

14. Peterson, W. W., *Error correcting codes*, MIT Press, (1961)

15. Pierce, W. H., *Fault tolerant computer design*, Academic Press, (1965)

16. Sellers, F. F. Jr., Hsiao, M. Y., Bearnson, L. W., *Error detecting logic for digital computers*, McGraw-Hill Book Company

17. Tryon, J. G., *Quadded logic, redundancy techniques for computing systems*, Spartan Books (1962)

18. Van Der Ziel, A., *Noise, sources, characterization, measurement* (1970)

19. Wilcox, R. H., Mann, W. C., *Redundancy techniques for computing systems*, Spartan Books (1962)

## b. ARTICLES AND REPORTS

20. Airinc Research Corporation, "A Technique For Evaluating Avionics Built-In Test, Final Report", Prepared for Naval Air Systems Command, Washington, D.C. Under Contract NC0019-71-C-0312, September (1971)

21. Armstrong, D. B., "On Finding a Nearly Minimal Set of Fault Detection Tests For Combinational Logic Nets", *IEEE Transactions on Electronic Computers*, EC-15, 60-73, February (1966)

22. Arnold, T. F., "The Concept of Coverage and Its Effect On The Reliability Model of a Repairable System", *IEEE TC Vol C-22*, March (1973)

23. Avizienis, A., "Arithmetic Error Codes: Cost and Effectiveness Studies for Application in Digital System Design", *Digest 1971 International Symposium On Fault Tolerant Computing*, IEEE Computer Society Publications. Northridge, California, pp 118-121

24. Avizienis, A., "Design of Fault-Tolerant Computers, *AFIPS Conference Proceedings*", Thompson Books, Washington, D.C., pp 733-743 (1967)

25. Avizienis et al, "The STAR (Self-Testing-And-Repairing) Computer: An Investigation of the Theory and Practice of Fault-Tolerant Computer Design", *IEE TC*, November C-20. No. 11, (1971)

26. Bearnson, L. W., Carroll, C. C., "On the Design of Minimum Length Fault Tests for Combinational Circuits", *Digest 1971 International Symposium On Fault Tolerant Computing*, IEEE Computer Society Publications, Northridge, California, pp 1-4

27. Benowitz, N., Calhoun, D. F., Alderson, G. E., Bauer, J. E., Joecker, C. T., "An Advanced Fault Isolation System for Digital Logic". *IEEE Transactions On Computers*, Vol. C-24, No. 5, May (1975)

28. Bielka, R. P., "Availability A System Function", *IRE Transactions on Reliability and Quality Control* RQC-9:38-42, September (1960)

29. Bossen, D. C., Ostapho, D. L., Patel, A. M., "Optimum Test Patterns for Parity Networks", *Computer Design*, Vol. 11, No. 4, April (1972)

30. Bouricius, W. G. et al, "Reliability Modeling For Fault Tolerant Computers' , *Digest 1971 International Symposium On Fault Tolerant Computing*, IEEE Computer Society Publications, Northridge, California

31. Bouricius, W. G., Carter, W. C., Schneider, P. R., "Reliability Modeling Techniques for Self-Repairing Computer Systems", *Proceedings ACM 1969 Annual Conference*, pp 295-309

32. Breuer, M. A., "Generation of Fault Detection Tests for Sequential Circuits", *Digest 1971 International Symposium on Fault Tolerant Computing*, IEEE Computer Society Publications, Northridge, California, pp 18-21

33. Carter, W. C., Schneider, P. R., "Design of Dynamically Checked Computers", *Proc. IFIPS* 68, North Holland, pp 878-883

34. Carter, W. C., Bouricius, W. G., Jessep, D. C., Roth, J. P., Schneider, P. R., Wadia, A. B., "A Theory of Design of Fault-Tolerant Computers Using Standby Sparing", *Digest of First International Symposium on Fault Tolerant Computing*, March (1971)

35. Carter, W. C., Duke, K. A., Jessep, D. C., "A Single Self-Testing Decoder Checking Circuit", *IEEE TC*, Vol. 20, November (1971)

36. Carter, W. C., Wadia, A. B., Jessep, D. C., "Implementation of Checkable Acyclic Automata By Morphic Boolean Functions", *Proc. Polytechnic Institute of Brooklyn on Computers and Automata*, March, (1972)

37. Carter, W. C., Jessep, D. C., Wadia, A. B., "Error-Free Decoding for Failure-Tolerance Memories", *Proceedings of IEEE International Computer Group Conference*, pp 229-239, June (1970)

38. Carter, W. C., Jessep, D. C., Wadia, A. B., Schneider, P. R., Bouricius, W. G., "Logic Design for Dynamic and Interactive Recovery", *IEEE TC*, Vol. C-20, No. 11, November (1971)

39. Carter, W. C., Jessep, D. C., Bouricius, W. C., Wadia, A. B., McCarthy, C. E., Milligan, F. G., "Design Techniques for Modular Architecture for Reliable Computer Systems", *Report No. RA12*, IBM Research, Yorktown Heights, March (1970) (Also NASA 70-208-000[illegible] NAST Tech. Report

40. Chang, H. Y., "An Algorithm for Selecting an Optimum Set of Diagnostic Tests", *IEEE Transactions on Electronic Computers*, Vol EC-14, No. 5, October (1965), pp 706-711

41. Chang, J. D., Kukel, J., "Advanced Torque Measurement System," Eustis Directorate U.S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, VA. *USAAMRDL Technical Report 73-54* August (1973)

42. Director, S. W., Rohrer, R. A., "The Generalized Adjoint Network Sensitivities" *IEEE Transactions on Circuit Theory*, August, 1969, pp 318-323

43. Falkoff, A. D., Iverson, K. E., "The APL Terminal System: Instructions for Operation," *IBM Watson Research Center*, Yorktown Heights. N.Y. February (1968)

44. Fantauzzi, G., Marsella, A., "Multiple-Fault Detection and Location in Fan-Out Free Combinational Circuits", *IEEE Transactions on Computers*, January (1974)

45. Friedman, A. D., "Fault Detection in Redundant Circuits," *IEEE Transactions on Electronic Computers,* EC-16(1), pp 99-100

46. Galey, S. M., Norby, R. E., Roth, J. P., "Techniques for Diagnosis of Switching Circuit Failures", *IEEE Transactions on Communications and Electronics* 33 (74), pp 509-514, (1964)

47. Grossman, S. E., "Instrument Makers Seek A Language". *Electronics,* September 19, 1974

48. Hogg, G. W., "Evaluation of the Effectiveness of Using Sonic Data to Diagnose the Mechanical Condition of Army Helicopter Power Train Components", U. S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va. *USAAMRDL Technical Report 72-30,* May (1972)

49. Hollinger, R. H., "Metallic Debris Monitor for Recirculating Lubricating Oil", U. S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va., *USAAMRDL Technical Report 72-10,* April (1972)

50. Hornbuckle, G. D. Spann, R. N., "Diagnosis of Single-Gate Failures in Combinational Circuits", *IEEE Transactions on Computers,* Vol C-18, No 3 March (1969) pp 216-220

51. Houser, D. R., Drosjack, M. J , "Vibration Signal Analysis Techniques," U.S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va., *USAAMRDL Technical Report 73-101,* December (1973)

52. Hennie, F. C , "Fault Detecting Experiments for Sequential Circuits", Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, Princeton, New Jersey, *IEEE Special Publications* S-164, November (1964) pp 95-110

53. Hsieh, E. P., "Checking Experiments for Sequential Machines," *IEEE Transactions on Computers,* Vol C-20. No. 10, October (1971), pp 1152-1165

54. Huffman, D. A., "The Synthesis of Sequential Switching Circuits", *Journal of the Franklin Institute,* March, (1954), Vol. 257. No 3

55. Jones, E. R., Mays, C. H., "Automatic Test Generation Methods for Large Scale Integrated Logic", *IEEE Journal of Solid State Circuits, SC-2, pp 221-226,* (1967)

56. Kautz, W. H., "Fault Testing and Diagnosis in Combinational Digital Circuits", *IEEE Transactions on Computers, EC-17(4), pp 352-366,* (1968)

57. Kohavi, Z., Lavalle, P., "Design of Diagnosable Sequential Machines with Fault-Detection Capabilities," *IEEE Transactions On Electronic Computers,* Vol EC-16, No 4, April, (1967) pp 473-484

58. Kos, J. M., Martin. A. J., Miller, P. D., et al, "Feasibility Investigation for determining Army Helicopter Gas Turbine Engine Maximum Power Available", U. S. Army Air Mobility Research and Development Laboratory, Ft Eustis, Va., *USAAMRDL Technical Report 72-58* (1972)

59. Landgraff, R. W., Yaw, S. S., "Design of Diagnosable Iterative Arrays", *IEEE Transactions on Computers,* Vol. C-20. No 8. pp 867-877. August (1971)

60. Loughry D. C., "Digital Bus Simplifies Instrument System Communication" *EDN*, September 1, 1972

61. Lynnworth, L. C., Pedersen, N. E., Carnevale, E. H., "Ultrasonic Mass Flowmeter for Army Aircraft Engine Diagnostics", U. S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va., *USAAMRDL Technical Report 72-66* (1972)

62. Manning, E., "On Computing Self-Diagnosis Part II – Generalizations and Design Principles", *IEEE Transactions on Electronic Computers*, Vol EC-15, No. 6, pp 882-890. December (1966)

63. Martin, R. R., "The Design of Diagnosable Sequential Machines", *Proceedings of the Hawaii International Conference on Computers* (1966)

64. Mauchly, J. W., "The Advantages of Built-In Checking", *Eastern Joint Computer Conference* (1953)

65. McKendry, J. M., Grant, G., Corso, J. F., "Assessing Maintainability", *Electronic Industries*, 19; 194-202 (1960)

66. McKevitt, J. F., "Parity Fault Detection in Semiconductor Memories", *Computer Design Vol. 11, No 7, pp 67-73*, July (1972)

67. Mealy, G. E., "A Method of Synthesizing Sequential Circuits", *Bell Systems Technical Journal*, 34, pp 1045-1079, (1955)

68. Moffatt, E. M., "High-Pressure Vibrating Pressure Transducer", U. S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va., *USAAMRDL Technical Report 72-42*, November (1972)

69. Moore, E. F., "Gedanken-Experiments on Sequential Machines," Automata Studies, Princeton University Press (1956)

70. Morant, C. D., "Automated Hybrid Circuit Testing", *Hybrid Microelectronic Symposium* (1958)

71. Moross, G. G., "Inductive Sensing Technique Advancement", U. S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va., *USAAMRDL Technical Report 71-51*, September (1971)

72. Nelson, G. E., Ricci, D. W., "Standard Instrument Simplifies System Design," *Electronics*, November 14, 1974

73. Norden Division, United Technologies, "Test Point Analysis & Self-Test Design Report for AGE Radar System, 3791R0074 Rev. F8, March 1973

74. Norden Division, United Technologies, "Technical Report Built-In Test for Radar System AN/APQ (92), 3672R1100."

75. Norden Division, United Technologies, "Determination of BIT Effectiveness Utilizing Simulation Language & General Purpose Computer," 3676R1101"

76. Norden Division, United Technologies, "Preliminary Outline of BIT Philosophy & Fault Isolation," 3676R1103

77. Von Neuman, J., "Probabilistic Logistic Logistics and the Synthesis of Reliable Organisms from Unreliable Components", *Automata Studies,* No 34. pp 43-98 (1956)

78. Poage, J. F., McCluskey, E. J., Jr., "Derivation of Optimum Test Sequences for Sequential Machines", Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, Princeton, N.J., *IEEE Special* Publication S-164, November (1964) pp 121-132

79. Ramamoorthy, C. V., Chang, L., "System Segmentation for the Parallel Diagnosis of Computers", *IEEE Transactions on Computers,* Vol. C-20, No. 3. pp 261-270, March (1971)

80. Ross, Fincock, Kavanagh. Wanamaker, "Automatic Test System for an Analog Computer", *IEEE Transactions on Instrumentation and Measurement,* Vol. IM-22, No 4, December (1973)

81. Roth, J. P., "Diagnosis of Automata Failures: A Calculus and A Method", *IBM Journal, Vol 10, No 4,* July (1966) pp 278-291

82. Roth, J. P., Bouricius, W. G., Schneider, P. R., "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits", *IEEE TEC.,* Vol. 16, No. 5 pp 567-579, October (1967)

83. Roth, J. P., Bouricius, W. G., Carter, W. C., Schneider, P. R., "Phase II of an Architectural Study for a Self-Repairing Computer", *SAMSO TR-67-106,* Nov (1967)

84. Sellers, F. F., Hsiao, M. Y., Bearnson, L. W., "Analyzing Errors with the Boolean Difference", *IEEE Transactions on Computers,* Vol C-17, No 7, pp 676-683, July (1968)

85. Szygenda, S. A., Flynn, M. J., "Self-Diagnosis and Self-Repair in Memory: An Integrated System Approach", *IEEE Transactions on Reliability,* Vol R-22, No 1, April (1973)

86. Szygenda, S. A., Goldbogen, "Implementation and Extension of Multi-Dimensional Path Sensitizing in a Simulation and Diagnosis System", *Proceedings of the Seventh Annual Allerton Conference on Circuit and System Theory,* October (1969)

87. Szygenda, S. A., Rouse, D. M., Thompson, E. W., "A Model and Implementation of a Universal Time Delay Simulator for Large Digital Nets", *AFIPS Conference Proceedings,* Vol. 34 (1970) SJCC, Thompson Books, Washington, D.C., pp 207-216

88. Szygenda, S. A., "A Software Diagnostic System for Test Generation and Simulation of Large Digital Systems", *Proceedings of the National Electronics Annual Conference,* Vol 25, December (1969)

89. Szygenda, S. A., "TEGAS-A Diagnostic Test Generation and Simulation System for Digital Computers", *Proceedings of the Third Hawaii International Conference on System Sciences,* January (1970)

90. Wakerly, J. F., "Partially Self-Checking Circuits and Their Use in Performing Logical Operations", *Digest 3rd International Symposium on Fault Tolerant Computing,* June (1973)

91. Writer, P., "Design for Testability, 1975", TETSO, Code 4050, NELC, San Diego

92. Yaw, S. S., Tang, Y. S., "An Efficient Algorithm for Generating Complete Test Sets for Combinational Logic Circuits", *Digest (1971) International Symposium On Fault-Tolerant Computing* IEEE Computer Society Publications, Northridge, California, pp 14-17

93. Ziebarth, H. K., Chang, J. D., Kukel, J., "Mechanical Component Failure Prognosis Study", U.S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Va. *USAAMRDL Technical Report 73-26,* June (1973)

## c. SPECIFICATIONS

94. DOD Directive 5000.1, Acquisition of Major Defense Systems.

95. Hughes, Ground Systems Group, application Note for the H4400 Computer.

96. Mil-E-5400, (ASG). Electronic Equipment Airborne, General Specification For.

97. Mil-E-16400, Military Specification, Electronic Equipment, Naval Ship and Shore, General Specification.

98. Mil-Handbook 217A, Reliability Stress and Failure Rate Data for Electronic Parts.

99. Mil-Handbook 239, (Navy), Military Standardization Handbook, Navy Standardization Program Application Handbook.

100. Mil-Handbook 472, Maintainability Prediction.

101. Mil-I-83287, Rev, C, Indicator, Fault Locating.

102. Mil-Std-415D, Test Provisions for Electronic Systems and Associated Equipment, Design Criteria For.

103. Mil-Std-471, Maintainability Demonstration.

104. Mil-Std-721B, Definition of Effectiveness Terms for Reliability, Maintainability, Human Factors, and Safety.

105. Mil-Std-756A, Military Standard, Reliability Prediction.

106. Mil-Std-781A, Reliability Tests Exponential Distribution.

107. Mil-Std-1309, Definition of Terms for Automatic Electronic Test and Checkout.

108. Mil-Std-1326, Test Points, Test Point Selection and Interface Requirements for Equipments Monitored by Shipboard On-Line Automatic Test Equipment.

109. Mil-Std-1390, Level of Repair.

110. NAVAIR 00-65 502, Handbook of Reliability Engineering.

111. NAVORD OD 39223, Maintainability Engineering Handbook.

112. NAVSHIPS 93820, Failure Calculations.

113. NAVSHIPS 94324, Maintainability Design Criteria Handbook for Design of Shipboard Electronic Equipment.

114. NAVTRADEVCEN 330-1-4, Maintainability Handbook for Electronic Equipment Design, Supplement IV. U.S. Naval Training Device Center, Ft. Washington, N.Y., April (1960), Technical Report Prepared by H. R. B. Singer, State College, Pa.

115. RADC-TR-69-356, Maintainability Prediction and Demonstration Techniques, Vol II, Rome Air Development Center, Report Nos. AD 869396 and AD 872873,-(TR-70-89).

116. WADC Technical Report 56-218, Guide to Design of Electronic Equipment for Maintainability, Wright Air Development Center, Wright-Patterson AFB, Ohio (1956)

117. Abbreviated Test Language for Avionics Systems (Atlas) Aeronautical Radio, 2551 Riva Rd., Annapolis, Md. 21401, Arinc Spec 416, March (1971)

118 General Requirements for Versatile Avionic Shop Test System/Avionics Compatability, Naval Air Systems Command, Requirement AR-8A.

119. General Requirements for Operational Test Program Sets, Naval Air Systems Command, Requirements AR-9A.

120. General Requirements for Maintainability of Avionics Equipment and Systems, Naval Air Systems Command, Requirement AR-10A.

121. Preparation of Test Requirements Document, Department of Air Force, Wright Patterson AFB Mil-Std-1519, Sept 17, 1971.

☆U.S. GOVERNMENT PRINTING OFFICE. 1979-614-02 /141

# *MISSION*
# *of*
# *Rome Air Development Center*

*RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence ($C^3I$) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.*